Analyse de données RADseq sous Galaxy : l'exemple de STACKS

Yvan Le Bras, Projet e-Biogenouest, CNRS UMR 6074 IRISA-INRIA, Rennes

Cyril Monjeaud, Projet GenoCloud, CNRS UMR 6074 IRISA-INRIA, Rennes

Avec l'utilisation de ressources réalisées par Julian Catchen, Institute of Ecology and Evolution, University of Oregon

Le but de ces exercices est de familiariser les stagiaires avec l'utilisation des données générées par les séquenceurs de nouvelles générations à partir de RRL (Reduced Representation Libraries) comme les tags associés à des sites de restriction (RAD). Ces librairies sont souvent utilisées dans le cadre du génotypage par séquençage, et peuvent fournir un jeu dense de marqueurs SNP (single nucleotide polymorphism) répartis le long du génome. Les stagiaires acquerront de l'expérience avec un pipeline d'analyse nommé STACKS, créé pour l'analyse de ce type de données. Les données à analyser proviendront d'un organisme sans génome de référence pour générer une cartographie génétique d'une part et identifier de potentielles signatures de sélection. Il est également possible d'utiliser un organisme avec génome de référence.

Les participants vont apprendre à:

1. Préparer les données brutes Illumina RAD pour leur analyse en enlevant les lectures de mauvaise qualité et pour démultiplexer un jeu d'échantillons barcodés.

2. Aligner des séquences RAD contre un génome de référence

3. Utiliser Stacks pour assembler les loci RAD, detecter des SNPs, les génotypes et haplotypes pour chaque individu de deux populations.

4. Calculer des statistiques en génétique des populations

A la fin de cette formation, vous devriez savoir:

5. Manipuler les données brutes Illumina de RAD pour les analyser en utilisant une variété de différents paramètres.

6. Aligner les tags RAD contre un génome de référence pour identifier des signatures potentielles de sélection.

7. Etendre ce qui a été appris vers des problèmes plus complexes, les vôtres.

Nous allons utiliser des jeux de données proposés par Julian dans ces formations plus des jeux de données épinoche de l'article d'Hohenlohe et al. 2010. Pour le nettoyage et l'analyse des données, nous nous reposerons principalement sur Galaxy, le pipeline STACKS et BWA.

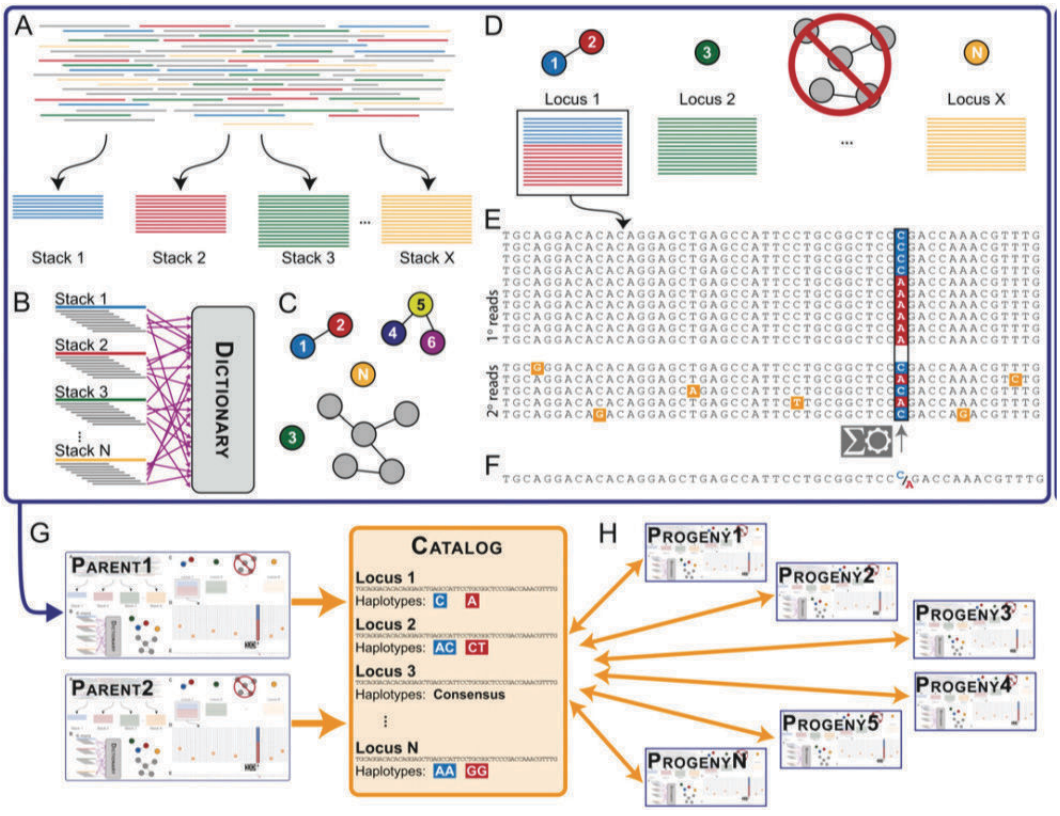
Les jeux de données seront tous produits via des séquenceurs de type Illumina GAII ou HiSeq2000.

Les logiciels sont tous open source

• ***BWA*** **(http://bio-bwa.sourceforge.net/)** - *BWA* est utilize pour aligner des séquences contre un génome de reference. Nous l'utiliserons pour aligner les lectures RAD contre le génome de l'épinoche, puis pour analyser ces lectures via le pipeline Stacks. Si nous utilisons BWA ici, plusieurs autres algorithmes et logiciels existent pour effectuer cette tâche (comme Bowtie) et ils peuvent également être utilisés pour cette partie. Nous avons cependant développé un outil spécialement pour STACKS sous Galaxy, basé sur BWA.

• ***Stacks*** **(http://creskoloab.uoregon.edu/stacks/)** – il s'agit d'un ensemble de programmes open source interconnectés initialement mis en place pour l'assemblage de novo de séquences RAD en loci et cartes génétiques, aujourd'hui étendu pour être utilisé de manière plus flexible dans des études d'organismes présentant ou non un génome de référence. Le pipeline a un wrapper Perl permettant le lancement de l'ensemble des programmes. La modularité de STACKS lui permet d'être appliqué à différents types de scénarios.

A noter que le pipeline STACKS utilisant un génome de référence suit les mêmes étapes. La principale différence s'observe pendant la phase de construction des loci. ***Ref_map*** construit alors les loci à partir du génome de référence, en utilisant les résultats de mapping des lectures sur ce dernier (réalisé via Bowtie, BWA ou un autre logiciel de mapping). Cette étape se nomme alors ***pstacks*** et se déroule à la place de ***usctacks***. ***cstacks*** et ***sstacks*** sont ensuite exécutés.



Il exécutera chaque composant de Stacks. En premier, il exécutera (A-F) ***ustacks*** sur chacun des échantillons, construisant les loci et détectant les SNPs à chacun d'entre eux. Les séquences présentant des correspondances exactes sont regroupées en piles ("stacks") (A). Un nombre de lectures trop faible dans une pile pouvant provenir d'erreur de séquençage, les piles uniques contenant moins de lectures que le seuil spécifié (*stack depth parameter*) sont désassemblés et les lectures mises de côté. Les lectures finalement conservées dans une pile sont nommées lectures

primaires. Celles mises de côté en raison d'un nombre de lectures trop faible par pile sont nommées lectures secondaires.

A la fin de cette première étape de création de piles, ***ustacks*** calcule la moyenne de profondeur de couverture et identifie les piles présentant une profondeur supérieure à 2 écarts types au-dessus de la moyenne. Toute les piles dans ce cas de figure (nommées piles bucheronnes), ainsi que celles présentant une séquence proche au nucléotide près, sont exclues car souvent représentées par des éléments répétés (voir cas de figure des 5 piles grises reliées (C et D)).

Ensuite, des sous-ensembles de piles sont produits (C), lorsque des piles sont très proches, un nucléotide de différence. Les piles de chaque sous-ensemble peuvent alors être réunis en un seul locus, comme c'est le cas pour les piles 1 et 2 (D). Ensuite, les lectures qui étaient initialement mises de côté, car présentant une similarité trop faible avec les séquences des piles (polymorphisme sur plus d'un nucléotide) (E), sont comparées à celles constitutives des piles pour identifier si elles peuvent être raccrocher à un unique polymorphisme identifié dans les étapes précédentes. Enfin, une séquence consensus est établie (F). En second, (G) ***cstacks*** sera exécuté pour créer un catalogue de tous les loci à partir des parents du croisement. Finalement, (H) ***sstacks*** s'exécutera pour évaluer la concordance entre les loci de chaque descendant et le catalogue de loci. Le pipeline identifie ensuite les loci représentant des marqueurs cartographiables (exécution du module genotypes).

En travaillant sur des données NGS, vous noterez probablement que toutes les données générées par le séquenceur ne sont pas de bonne qualité. En général, il faudra enlever les séquences de faibles qualités de vos jeux de données avant de les analyser. En même temps, la stringence de la filtration dépendra de l'application finale. En général, une stringence plus forte est appliquée pour un assemblage de novo comparé à l'utilisation d'alignements contre un génome de référence. Par contre, des données de mauvaise qualité affecteront presque toujours les analyses futures, en produisant des faux positifs, comme par exemple la prédiction de faux SNP.

I. Analyse RAD-seq sous Galaxy : Détection de SNPs

1. L'analyse

Nous travaillerons sur les données issues de l'archive pe_sample.tar.gz disponible sur le site de Stacks (http://creskolab.uoregon.edu/stacks/pe_tutorial/pe_samples.tar.gz). Sous Galaxy, commencez par créer un nouvel historique (ex: "*RAD 1 : SNP calling*").

Récupération des données brutes dans Shared data/data libraries/1 Galaxy teaching folder/2013_GenOuest2/genetic_map

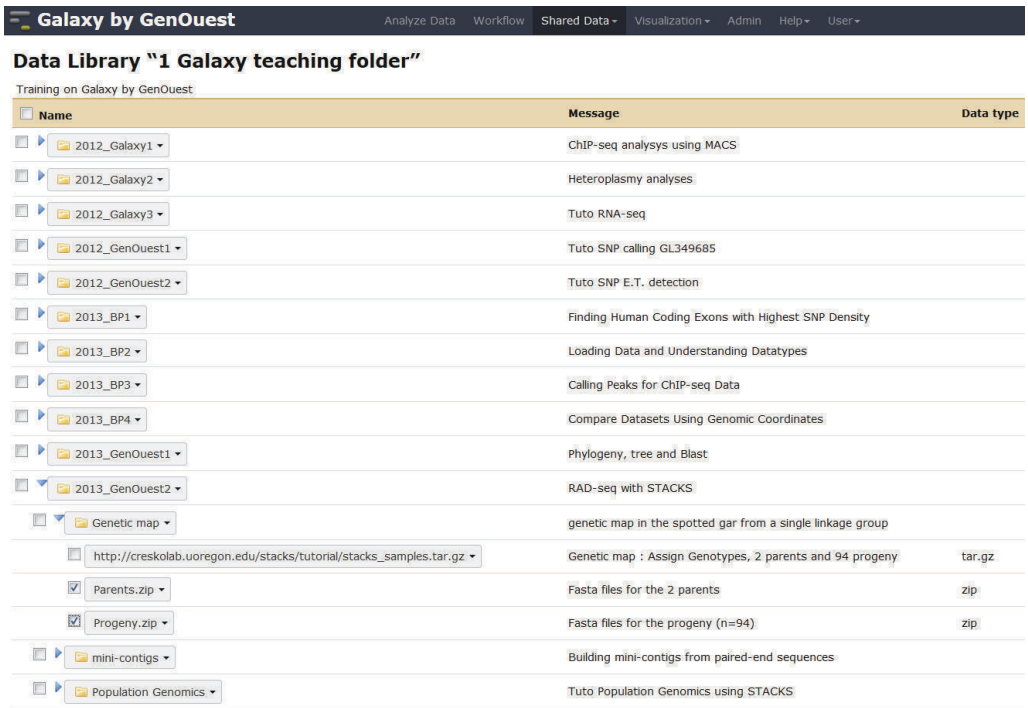
Vous y trouverez 3 archives :

-une contenant les 95 jeux de données de séquences : stacks_samples.tar.gz

-une regroupant les 2 jeux de données de séquences des parents : Parents.zip

-une regroupant les 93 jeux de données de séquences des descendants F1 : Progeny.zip

Sélectionner les 2 dernières archives .zip (***Parents*** et ***Progeny***), et cliquer sur le bouton "*GO*" après avoir vérifié que l'action sélectionnée était bien "*Import to current history*".



Vous vous retrouvez avec un historique contenant les deux archives au format zip.

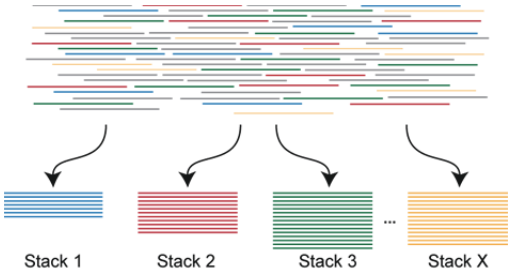
Les données ayant déjà été nettoyées et démultiplexées, il ne sera pas nécessaire d'exécuter ***STACKS : Process Radtags***.

Nous sommes maintenant près à exécuter le pipeline ***STACKS denovo***.

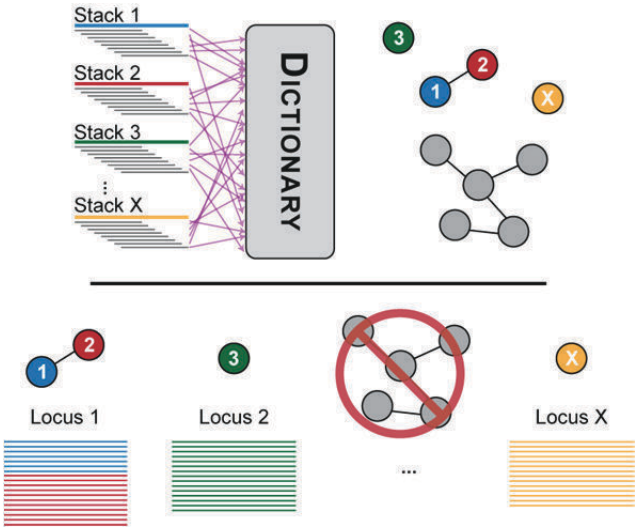
Exécuter ***STACKS : de novo map*** sur les individus étudiés (par exemple les parents) en prenant comme input l'archive .zip et en spécifiant l'usage "*Genetic map*". Décocher l'option "*create zip outputs*"

Vous pouvez préciser certains paramètres. Ainsi, en sélectionnant le mode Advanced pour le paramètre ***Stack assembly options***, vous pouvez préciser:

-la profondeur minimum de la pile (-m), ici "Minimum number of identical". Ce paramètre, passé à ***ustacks***, contrôle le nombre de lectures avec une correspondance exacte devant être trouvé pour créer une pile chez un individu. Nous pouvons sélectionner 3, une pile ne sera générée chez un individu que si au moins 3 lectures correspondent exactement. Les lectures alors utilisées pour constituer ces piles sont qualifiées de lectures primaires. Les autres, qui ne correspondent donc pas exactement à des lectures primaires et qui n'étaient pas assez nombreuses pour générer une pile, sont qualifiées de lectures secondaires.



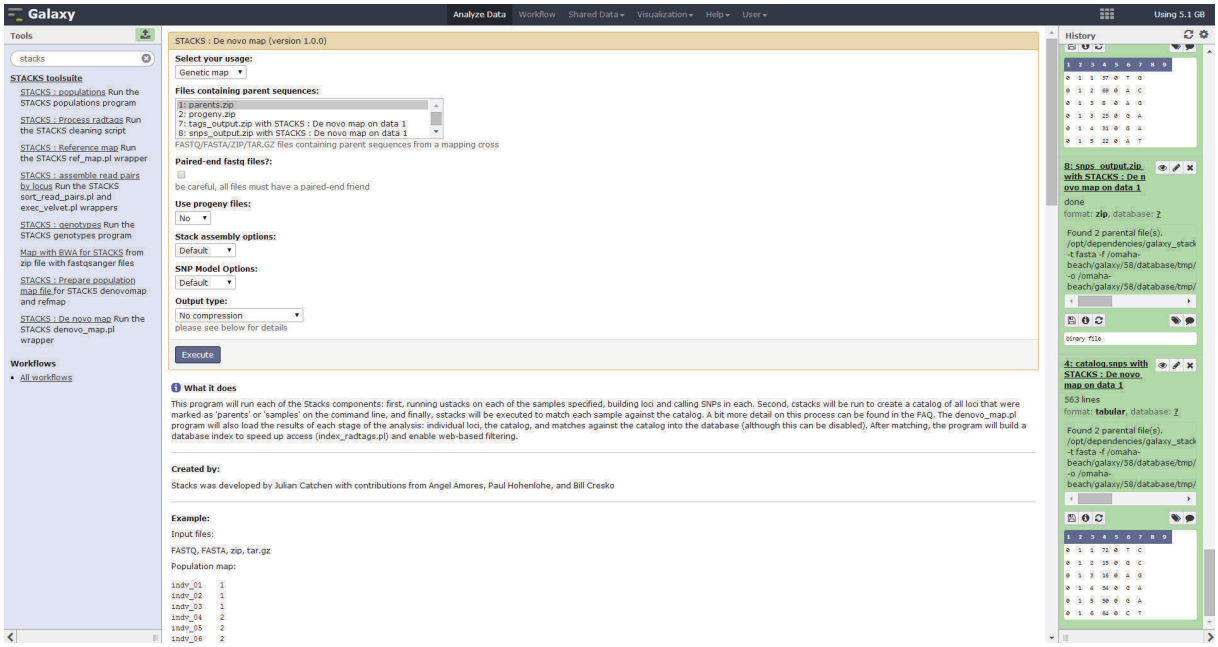
-la distance maximum permise entre des piles pour qu'elles soient fusionnées en un locus potentiel chez un individu. Il s'agit du paramètre –M, ici "Number of mismatches allowed between loci when processing a single individual". Nous préciserons ici une distance maximale de 2 nucléotides. En reprenant le schéma ci-dessous, les piles 1 et 2 vont être fusionnées car représentent un seul locus polymorphique (avec 2 allèles différents possédant une différence maximale correspondante au seuil fixé, ici 2), les piles 3 et X vont être associés à deux loci monomorphiques distincts, le gros paquet gris de piles représente un jeu de séquences répétées qui présentent trop d'allèles pour être biologiquement corrects.



-le nombre maximal de différence entre loci du catalogue. Il s'agit du paramètre –n, "specify the number of mismatches allowed between loci when building the catalog" ici. Ce paramètre permet notamment de pouvoir créer un locus de type homozygote dans le catalogue alors qu'il est en réalité hétérozygote. Ainsi, si par exemple on fixe la valeur à 2 et que la séquence du locus ne présente pas de polymorphisme au sein de chaque individu mais présente plus de 2 nucléotides de différences entre chaque individu, alors chaque "version" du locus identifié générera un locus distinct des autres. Ce paramètre est notamment pratique pour conserver les loci hétérozygotes entre parents mais homozygote chez chacun d'entre eux. Si la valeur de ce paramètre est trop importante, cstacks risque de considérer des loci distincts en un seul locus. Si cette valeur est trop faible, cstacks générera potentiellement plus de loci qu'il n'y en a réellement. Nous pouvons fixer ce paramètre à 3.

-supprimer ou casser les RAD-tags très répétitifs. Il s'agit de supprimer les piles "bûcheronnes" de l'analyse et briser les piles modérément surdimensionnées. Nous cocherons ici cette option, correspondant au paramètre -t.

D'une manière générale, utiliser des options plus stringentes (comme la suppression des RAD-tags très répétitifs) va générer un nombre total de piles plus faible, augmentera le nombre de piles réunis (car manque des piles à n nucléotides près faisant le pont entre deux piles) et diminuera le nombre final de loci du catalogue. (car moins de similarité, plus de bruit de fond, entre les piles des individus).



Ne pas oublier de rafraîchir l'historique une fois le job terminé pour faire apparaître les jeux de données de sorties additionnels!

2. Les fichiers de sortie

Soit le pipeline lancé sur deux individus, "male" et "female". Dans les fichiers générés, nous retrouvons:

a) Un premier jeu de données nommé result.log

Il permet notamment de vérifier le bon déroulement du job.

```
denovo_map.pl started at 2014-02-14 12:37:49
/local/galaxy/stacks-1.09/bin/denovo_map.pl -p /
```

Nous pouvons voir tout d'abord le script lancé (ici denovo_map.pl), ainsi que la date et l'heure de départ du job (ici le 14 février 2014 à midi 37, bonne saint Valentin ;)). Ensuite, vient la ligne de commande lancée, `/local/galaxy/stacks-1.09/bin/denovo_map.pl -p…`
S'en suit le listing des étapes effectuées fichier par fichier :

```
Identifying unique stacks; file   1 of   2 [female]
/opt/dependencies/galaxy_stacks/1.1.0/cmonjeau/stacks_dependencies/2ea494032b10/bin/ustacks -t fasta -f /omaha-
beach/galaxy/58/database/tmp/tmpNo7qXs/tmpqVMSAn/female.fa -o /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S -i 1  -p 4 2>&1
Min depth of coverage to create a stack: 3
Max distance allowed between stacks: 2
Max distance allowed to align secondary reads: 4
Max number of stacks allowed per de novo locus: 3
Deleveraging algorithm: disabled
Removal algorithm: disabled
Model type: SNP
Alpha significance level for model: 0.05
Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpqVMSAn/female.fa
Loaded 29517 RAD-Tags; inserted 4922 elements into the RAD-Tags hash map.
  0 reads contained uncalled nucleotides that were modified.
  Mean coverage depth is 35; Std Dev: 22.8266 Max: 150
Coverage mean: 35; stdev: 22.8266
Deleveraging trigger: 58; Removal trigger: 81
Calculating distance between stacks...
  Distance allowed between stacks: 2
  Using a k-mer length of 25
  Number of kmers per sequence: 51
  Miniumum number of k-mers to define a match: 1
Merging stacks, maximum allowed distance: 2 nucleotide(s)
  715 stacks merged into 450 stacks; deleveraged 0 stacks; removed 1 stacks.
  Mean merged coverage depth is 55.8911; Std Dev: 29.1947; Max: 195
Merging remainder radtags
  4366 remainder sequences left to merge.
  Distance allowed between stacks: 4
  Using a k-mer length of 15
  Number of kmers per sequence: 61
  Miniumum number of k-mers to define a match: 1
  Matched 4346 remainder reads; unable to match 20 remainder reads.
Number of utilized reads: 29497
Writing results
  Refetching sequencing IDs from /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpqVMSAn/female.fa... read 29517 sequence IDs.
```

Nous obtenons ainsi, pour le premier des 2 fichiers, le parent "female" :

-la ligne de commande correspondant au lancement de ***ustacks***

-les paramètres renseignés lors de la soumission du job (Min depth of coverage, ….)

-le nombre total de lectures présentant la signature d'un site de restriction, qualifiées de RAD-tags, chargées (ici 29517) ainsi que le nombre d'éléments insérés dans la carte de hachage de RAD-tags (ici 4922).

-le nombre de lectures avec des nucléotides modifiés

-la profondeur de couverture moyenne

-le nombre de stacks fusionnés car proches (ici 715 fusionnés en 450).

-le nombre de lectures restantes concordant avec les 450 stacks préalablement générés. Ici 4346.

Sur les 29517 lectures d'origines "RAD tagguées", seules 20 n'ont pas été utilisées. Il y a donc 29497 lectures finalement utilisées pour générer 450 stacks.

Puis ***ustacks*** est exécuté sur le parent "male"

Ensuite est exécuté ***cstacks***.

```
/opt/dependencies/galaxy_stacks/1.1.0/cmonjeau/stacks_dependencies/2ea494032b10/bin/cstacks -b 1 -o /omaha-
beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S -s /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female -s /omaha-
beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male  -p 4 2>&1
Number of mismatches allowed between stacks: 0
Loci matched based on sequence identity.
Constructing catalog from 2 samples.
Initializing new catalog...
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female.tags.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female.snps.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female.alleles.tsv
Processing sample 2
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male.tags.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male.snps.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male.alleles.tsv
Searching for sequence matches...
  Number of kmers per sequence: 1
  Miniumum number of k-mers to define a match: 1
  449 loci in the catalog, 678 kmers in the catalog hash.
Merging matches into catalog...
  0 loci matched more than one catalog locus and were excluded.
Writing catalog...
```

Là encore, un rappel des paramètres utilisés est fait. Trois types de fichiers différents générés par ***ustacks*** sont utilisé par individu, à savoir *tags.tsv*, *snps.tsv* et *alleles.tsv*. Nous reviendrons sur ces fichiers dans les sections suivantes. Le catalogue de loci est ensuite créé à partir des échantillons parentaux. Ici, le catalogue créé contient 462 loci (voir *catalog.tags)* dont 449 provenant de l'"individu de référence" (female) et pouvant être partagé avec le second individu (male).

Enfin, ***sstacks*** est exécuté:

```
/opt/dependencies/galaxy_stacks/1.1.0/cmonjeau/stacks_dependencies/2ea494032b10/bin/sstacks -b 1 -c /omaha-
beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1 -s /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female -o /omaha-
beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S -p 4 2>&1
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.tags.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.snps.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.alleles.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female.tags.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female.snps.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female.alleles.tsv
Searching for sequence matches...
449 stacks compared against the catalog containing 462 loci.
  449 matching loci, 0 contained no verified haplotypes.
  0 loci matched more than one catalog locus and were excluded.
  0 loci contained SNPs unaccounted for in the catalog and were excluded.
  678 total haplotypes examined from matching loci, 678 verified.
Outputing to file /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female.matches.tsv
/opt/dependencies/galaxy_stacks/1.1.0/cmonjeau/stacks_dependencies/2ea494032b10/bin/sstacks -b 1 -c /omaha-
beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1 -s /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male -o /omaha-
beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S -p 4 2>&1
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.tags.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.snps.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.alleles.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male.tags.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male.snps.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male.alleles.tsv
Searching for sequence matches...
457 stacks compared against the catalog containing 462 loci.
  457 matching loci, 0 contained no verified haplotypes.
  0 loci matched more than one catalog locus and were excluded.
  0 loci contained SNPs unaccounted for in the catalog and were excluded.
  732 total haplotypes examined from matching loci, 732 verified.
Outputing to file /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male.matches.tsv
/opt/dependencies/galaxy_stacks/1.1.0/cmonjeau/stacks_dependencies/2ea494032b10/bin/genotypes -b 1 -P /omaha-
beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S -r 1 -c -s  2>&1
Found 2 input file(s).
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.tags.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.snps.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.catalog.alleles.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/female.matches.tsv
  Parsing /omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/male.matches.tsv
Identified parent IDs: 1 2
Populating observed haplotypes for 2 samples, 462 loci.
Performing automated corrections...
0 potential genotypes in 460 markers, 0 populated; 0 corrected, 0 converted to heterozygotes, 0 unsupported homozygotes removed.
Writing 443 loci to genotype file, '/omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.genotypes_1.tsv'
Writing SQL markers file to '/omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.markers.tsv'
Writing SQL genotypes file to '/omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.genotypes_1.txt'
Writing 462 loci to observed haplotype file, '/omaha-beach/galaxy/58/database/tmp/tmpNo7qXs/tmpAChs9S/batch_1.haplotypes_1.tsv'
denovo_map.pl completed at 2014-08-07 12:03:51
```

Pour chaque individu, ***sstacks*** détermine l'haplotype à chaque locus de chacun des individus du croisement. Il va alors comparer les stacks de chaque individu (female : 449 loci / male : 457 loci) avec le catalogue complet contenant 462 loci. Ces 462 loci sont ceux du fichier *catalog.tags.*

En parcourant ce fichier, identifiez les loci spécifiques à chacun des individus et ceux partagés

Ces 462 loci représentent les 444 de l'individu "female" partagés avec l'individu "male" + 5 loci spécifiques à l'individu "female" + 13 loci spécifique à l'individu "male"). Ici, 678 haplotypes sont trouvés au total pour l'individu "female", 732 pour "male".

b) Les fichiers tags

XXX.tags.tsv file:

Column	Name	Description
1	Sql ID	This field will always be "0", however the MySQL database will assign an ID when it is loaded.
2	Sample ID	Each sample passed through Stacks gets a unique id for that sample.
3	Stack ID	Each stack formed gets an ID.
4	Chromosome	If aligned to a reference genome using pstacks, otherwise it is blank.
5	Basepair	If aligned to ref genome using pstacks.
6	Strand	If aligned to ref genome using pstacks.
7	Sequence Type	Either 'consensus', 'primary' or 'secondary', see the Stacks paper for definitions of these terms.
8	Sequence ID	The individual sequence read that was merged into this stack.
9	Sequence	The raw sequencing read.
10	Deleveraged Flag	If "1", this stack was processed by the deleveraging algorithm and was broken down from a larger stack.
11	Blacklisted Flag	If "1", this stack was still confounded depsite processing by the deleveraging algorithm.
12	Lumberja ckstack Flag	If "1", this stack was set aside due to having an extreme depth of coverage.

Notes: For the tags file, each stack will start in the file with a consensus sequence for the entire stack followed by the flags for that stack. Then, each individual read that was merged into that stack will follow. The next stack will start with another consensus sequence.

Exemple du *catalog.tags.tsv* :

#Sql_ID	Sample_ID	Stack_ID	Chromosome	Basepair	Strand	Sequence_Type	Allele	Sequence_ID	Sequence	Deleveraged_Flag	Blacklisted_Flag	Lumberja_ckstack_Flag
0	1	1		0	+	consensus	0	1_1,2_5	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG	0	0	0
0	1	2		0	+	consensus	0	1_2,2_199	TGCAGGAGCGTCTTAGACCTCCCCTGTCTTAGGCCTAGAAGATCATCGAATAATGTCCCTTGCATTGTGGCCAAA	0	0	0
0	1	3		0	+	consensus	0	1_3,2_16	TGCAGGTTTCATGGTGGTGTCTGTACCACTGTCTTACTGGATTGTGTGGAAGAACTGTAGGATCTCATGCTGGTG	0	0	0
0	1	4		0	+	consensus	0	1_4,2_176	TGCAGGCTAACATGCAGGAGGACAGAGCAAACACATACTGTGTGCAATACAGTAGTTGCAGTTTGTATTATTTCA	0	0	0
0	1	5		0	+	consensus	0	1_5,2_171	TGCAGGCCAGCATCGATGAAGACAGTCAAAGGTGAGGAGGCGAGGACCCAGAGGTGTCTCGGTCCCGATCCAGTG	0	0	0
0	1	6		0	+	consensus	0	1_6,2_267	TGCAGGCCTTCTGAAGCTGCTGACTCAGTGTTTATTGATTAGCACATGCACAATTAATTTTAAGCCTTGTCTTCT	0	0	0
0	1	7		0	+	consensus	0	1_7,2_10	TGCAGGAGGTGTTGTTCTTCAGACAAGACATTAGACGGAGGTCCTGACGTCTTGTTTAACTAAAGATCCTGTATT	0	0	0
0	1	8		0	+	consensus	0	1_8,2_371	TGCAGGTTATGAAATTTAAGGAATAGCAAAGGCTATAAAAATCCAGTAACCTCTTCTAATTCATATAATTAAATG	0	0	0
0	1	9		0	+	consensus	0	1_9,2_334	TGCAGGCAGGAGAACCGTCAGACCCATGTTTCCCCTGTAGACAGAACTAACTCAGGTGGGTCTGTTTGCCAGTAA	0	0	0
0	1	10		0	+	consensus	0	1_10,2_14	TGCAGGGAACACAATCCCAGACAAACTGGGAGGGATCAGAGAGTTCTGCCTGTAGAAGTGCAGACAGGGCAAAGA	0	0	0
0	1	11		0	+	consensus	0	1_11,2_58	TGCAGGCCTCTGGGGCTCAAAATCTTTTTAATTTGTCAGCACAAGGTTAGCTTCTGCTTGCTTCTTAGGAGAAAA	0	0	0
0	1	12		0	+	consensus	0	1_12,2_36	TGCAGGACCAGGACTGGACAGCCATGTGGGCCTTCGTCACCCCGCGGTTTTGTACAGAGCTATCAAAAACGTCAG	0	0	0
0	1	13		0	+	consensus	0	1_13,2_15	TGCAGGGGTGCATATTACCCTGTCTCTGTAAAATGGAAGAGATATCACAGTGTTCAGTGAGGCATGCAGTTTATG	0	0	0
0	1	14		0	+	consensus	0	1_14,2_13	TGCAGGTCATACATAGTTCAATAAATTTATGTTTTAACAAAAACATGTTTTTCCACTAGAAATATCAAAATAGAA	0	0	0
0	1	15		0	+	consensus	0	1_15,2_17	TGCAGGTAACAGGAGATTTATTGAGTTTCCTTTAGTGTGCAACAGCAATGCACCACATGTTTCGGTGCTAGATGT	0	0	0
0	1	16		0	+	consensus	0	1_16,2_265	TGCAGGACTCATTTTAAATTCTGCAGCCCCTTCTACCATGTTTGCCATTTTATCAATATCAGAAGGCACACAGGT	0	0	0
0	1	17		0	+	consensus	0	1_17,2_387	TGCAGGTGATTATACCTTCCGTACTGCTCACACTGCACACTGTATCACACACTGCAGCATGCACACTGGTCAGTC	0	0	0

Nous pouvons constater que l'identifiant "*Sequence_ID*" est constitué à partir des ID des individus et des piles correspondantes comme suit : "*individu1_stackIDindividu1, individu2_stackIDindividu2*". Il est ainsi directement aisé de savoir quels SNPs sont détectés sur des loci séquencés chez nos 2 individus.

Pour un *sample.tags.tsv* :

#SQL_ID	Sample_ID	Stack_ID	Chromosome	Basepair	Strand	Sequence_Type	Allele	Sequence_ID	Sequence	Deleveraged_Flag	Blacklisted_Flag	Lumberja_ckstack_Flag
0	1	1		0	+	consensus			TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG	0	0	0
0	1	1				model			OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOEOO			
0	1	1				primary	0	GGAAG_2_0037_1653_1588_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	GGAAG_2_0056_1218_1292_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	GGAAG_2_0059_413_117_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	GGAAG_2_0059_429_168_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	GGAAG_2_0095_418_269_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CTAGG_2_0007_1667_1559_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CTAGG_2_0019_1041_1863_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CTAGG_2_0056_1118_819_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CTAGG_2_0067_495_1323_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CTAGG_2_0092_192_532_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CTAGG_2_0097_1609_585_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CTAGG_2_0100_319_709_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0002_1519_640_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0009_1752_1558_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0013_1167_1545_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0016_263_1281_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0019_1063_739_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0021_1543_547_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0028_130_956_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0038_1052_772_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0038_1118_279_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0041_1195_1379_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0045_172_639_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0046_225_405_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0051_674_1617_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0055_1597_494_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0056_1677_815_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0057_178_1779_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0062_13_851_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0062_800_610_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0067_561_1807_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0069_1473_640_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0072_1026_1898_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0078_970_289_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0087_1633_1356_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	0	CAGTC_2_0096_641_629_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG			
0	1	1				primary	1	GGAAG_2_0012_1275_1694_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG			
0	1	1				primary	1	GGAAG_2_0019_481_1131_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG			
0	1	1				primary	1	GGAAG_2_0035_626_1637_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG			
0	1	1				primary	1	GGAAG_2_0062_618_273_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG			
0	1	1				primary	1	GGAAG_2_0090_749_1499_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG			

Primary vs secondary reads :

0	1	1	primary	1	CAGTC_2_0076_1442_1576_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG
0	1	1	primary	1	CAGTC_2_0079_1540_762_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG
0	1	1	primary	1	CAGTC_2_0080_1139_865_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG
0	1	1	primary	1	CAGTC_2_0081_458_1052_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG
0	1	1	primary	1	CAGTC_2_0088_1542_1441_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG
0	1	1	primary	1	CAGTC_2_0092_629_1368_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG
0	1	1	primary	1	CAGTC_2_0093_1732_263_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG
0	1	1	primary	1	CAGTC_2_0097_697_1869_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGG
0	1	1	secondary		CAGTC_2_0082_1602_271_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATATGTTTTGTGTCTGCATGG
0	1	1	secondary		CAGTC_2_0093_92_1898_1[79984]	TGCAGGTACCTCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG
0	1	1	secondary		CAGTC_2_0058_1624_135_1[79984]	TGCAGGTACATCAATCAATCGGACTACACCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG
0	1	1	secondary		CAGTC_2_0091_1720_1032_1[79984]	TGCAGGTACATCAATCAAACGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCATGG
0	1	1	secondary		CTAGG_2_0038_1473_926_1[79984]	TGCAGGTACATCAATCAATCGGACTACATCTGAACCACCTGATCCAACAAAACTATGTGTTTTGTGTCTGCACGA

Utiliser les fichiers individu.tags.tsv pour retrouver le nombre de tags par individu. Vous serez probablement amenés à utiliser l'outil "count"

c) Les fichiers alleles

```
Column    Name                     Description
1         Sql ID                   This field will always be "0", however the MySQL
database will assign an ID when it is loaded.
2         Sample ID
3         Stack ID
4         Haplotype                The haplotype, as constructed from the called
SNPs at each locus.
5         Percent                  Percentage of reads that have this haplotype
6         Count                    Raw number of reads that have this haplotype
```

Sample1.alleles.tsv à gauche et *catalog.alleles.tsv* à droite

#SQL_ID	Sample_ID	Stack_ID	Haplotype	Percent	Count
0	1	1	C	48.0519	37
0	1	1	T	51.9481	40
0	1	7	C	45.283	24
0	1	7	G	54.717	29
0	1	9	A	40.7407	22
0	1	9	G	59.2593	32
0	1	13	C	44.1558	34
0	1	13	T	55.8442	43
0	1	14	G	56.25	36
0	1	14	T	43.75	28
0	1	15	C	48.3871	75
0	1	15	T	51.6129	80
0	1	18	A	46.7742	29
0	1	18	T	53.2258	33
0	1	21	C	59.7403	46
0	1	21	T	40.2597	31
0	1	22	G	62.5	35
0	1	22	T	37.5	21
0	1	25	A	47.6923	31
0	1	25	C	52.3077	34
0	1	26	C	66.6667	16

#SQL_ID	Sample_ID	Stack_ID	Haplotype	Percent	Count
0	1	1	C	0	0
0	1	1	T	0	0
0	1	2	C	0	0
0	1	2	G	0	0
0	1	3	A	0	0
0	1	3	G	0	0
0	1	4	A	0	0
0	1	4	G	0	0
0	1	5	A	0	0
0	1	5	G	0	0
0	1	6	C	0	0
0	1	6	T	0	0
0	1	7	C	0	0
0	1	7	G	0	0
0	1	8	GC	0	0
0	1	8	TA	0	0
0	1	9	A	0	0
0	1	9	G	0	0
0	1	10	C	0	0
0	1	10	G	0	0
0	1	11	C	0	0

Sample2.alleles.tsv

#SQL_ID	Sample_ID	Stack_ID	Haplotype	Percent	Count
0	2	6	AT	42.3077	11
0	2	6	GC	57.6923	15
0	2	7	C	56.6265	47
0	2	7	T	42.1687	35
0	2	11	C	57.1429	48
0	2	11	T	41.6667	35
0	2	14	C	39.726	29
0	2	14	G	60.274	44
0	2	16	A	72.8571	51
0	2	16	G	27.1429	19
0	2	20	A	59.2593	16
0	2	20	C	40.7407	11
0	2	23	G	47.6923	31
0	2	23	T	52.3077	34
0	2	26	CG	60.5634	43
0	2	26	TA	39.4366	28
0	2	30	C	60	45
0	2	30	T	40	30
0	2	32	A	43.9024	36
0	2	32	G	56.0976	46
0	2	33	A	58.7156	64

Nous pouvons alors constater que si le Stack_ID 5 présente un SNP dans le fichier catalog.alleles.tsv, il n'est retrouvé dans aucun des fichiers samples (1 ou 2)…..

C'est parce que le Stack_ID du sample 1 ne correspond pas au Stack_ID du sample 2 qui lui-même ne correspond pas au Stack_ID du catalog…

Pour s'en rendre compte, il suffit de consulter les fichiers *sample**.matches.tsv*

d) Les fichiers matches

```
Column     Name                      Description
1          Sql ID                    This field will always be "0", however the MySQL
database will assign an ID when it is loaded.
2          Batch ID
3          Catalog ID
4          Sample ID
5          Stack ID
6          Haplotype
7          Stack Depth
```

Sample1.tsv à gauche et *sample2.alleles.tsv* à droite

#SQL_ID	Batch_ID	Catalog_ID	Sample_ID	Stack_ID	Haplotype	Stack_depth
0	1	1	1	1	C	37
0	1	1	1	1	T	40
0	1	2	1	2	G	47
0	1	3	1	3	G	33
0	1	4	1	4	A	53
0	1	5	1	5	G	21
0	1	6	1	6	C	73
0	1	7	1	7	C	24
0	1	7	1	7	G	29
0	1	8	1	8	TA	50
0	1	9	1	9	A	22
0	1	9	1	9	G	32
0	1	10	1	10	C	67
0	1	11	1	11	C	39
0	1	12	1	12	G	49
0	1	13	1	13	C	34
0	1	13	1	13	T	43
0	1	14	1	14	G	36
0	1	14	1	14	T	28
0	1	15	1	15	C	75
0	1	15	1	15	T	80

#SQL_ID	Batch_ID	Catalog_ID	Sample_ID	Stack_ID	Haplotype	Stack_depth
0	1	18	2	1	T	95
0	1	22	2	2	T	58
0	1	315	2	3	T	81
0	1	391	2	4	T	75
0	1	1	2	5	T	92
0	1	227	2	6	AT	11
0	1	227	2	6	GC	15
0	1	388	2	7	AC	47
0	1	388	2	7	AT	35
0	1	259	2	8	T	48
0	1	274	2	9	G	42
0	1	7	2	10	G	63
0	1	139	2	11	C	48
0	1	139	2	11	T	35
0	1	44	2	12	AA	32
0	1	14	2	13	G	70
0	1	10	2	14	C	29
0	1	10	2	14	G	44
0	1	13	2	15	T	45
0	1	3	2	16	A	51
0	1	3	2	16	G	19

Le Catalog_ID (= Stack_ID du catalog), reprend le Stack_ID de l'individu de "référence", ici sample 1, mais la numérotation est bien différente de celle du Stack_ID du sample 2…. Ainsi, dans le fichier "*catalog.alleles.tsv*", le Stack_ID 3 correspond au Stack_ID 16 du sample 2!

e) Les fichiers snps

```
Column     Name                    Description
1          Sql ID                  This field will always be "0", however the MySQL
database will assign an ID when it is loaded.
2          Sample ID
3          Stack ID
4          SNP Column
5          Likelihood ratio        From the SNP-calling model.
6          Rank_1                  Majority nucleotide.
7          Rank_2                  Alternative nucleotide.
```

Notes: If a stack has two SNPs called within it, then there will be two lines in this file listing each one.

sample.snps.tsv à gauche et *catalog.snps.tsv* à droite

#SQL_ID	Sample_ID	Stack_ID	SNP_Column	Likelihood_ratio	Maj_nt	Alt_nt
0	1	1	72	-81.1804	T	C
0	1	7	21	-52.261	G	C
0	1	9	16	-46.4764	G	A
0	1	13	70	-73.6513	T	C
0	1	14	58	-60.5197	G	T
0	1	15	20	-164.631	T	C
0	1	18	28	-63.4613	T	A
0	1	21	72	-65.1731	C	T
0	1	22	53	-42.6043	G	T
0	1	25	69	-67.9754	C	A
0	1	26	36	-14.8594	C	T
0	1	30	46	-76.5693	C	T
0	1	31	38	-77.3799	A	T
0	1	33	48	-76.7795	G	T
0	1	36	73	-59.2943	G	A

#SQL_ID	Sample_ID	Stack_ID	SNP_Column	Likelihood_ratio	Maj_nt	Alt_nt
0	1	1	72	0	T	C
0	1	2	15	0	G	C
0	1	3	16	0	A	G
0	1	4	34	0	G	A
0	1	5	50	0	G	A
0	1	6	64	0	C	T
0	1	7	21	0	G	C
0	1	8	15	0	G	T
0	1	8	66	0	C	A
0	1	9	16	0	G	A
0	1	10	36	0	G	C
0	1	11	23	0	G	C
0	1	12	43	0	G	A
0	1	13	70	0	T	C
0	1	14	58	0	G	T

Nous avons détecter des SNPs chez nos 2 individus et nous pouvons déterminer lesquels sont situés sur les même loci.

3. Retour à l'analyse

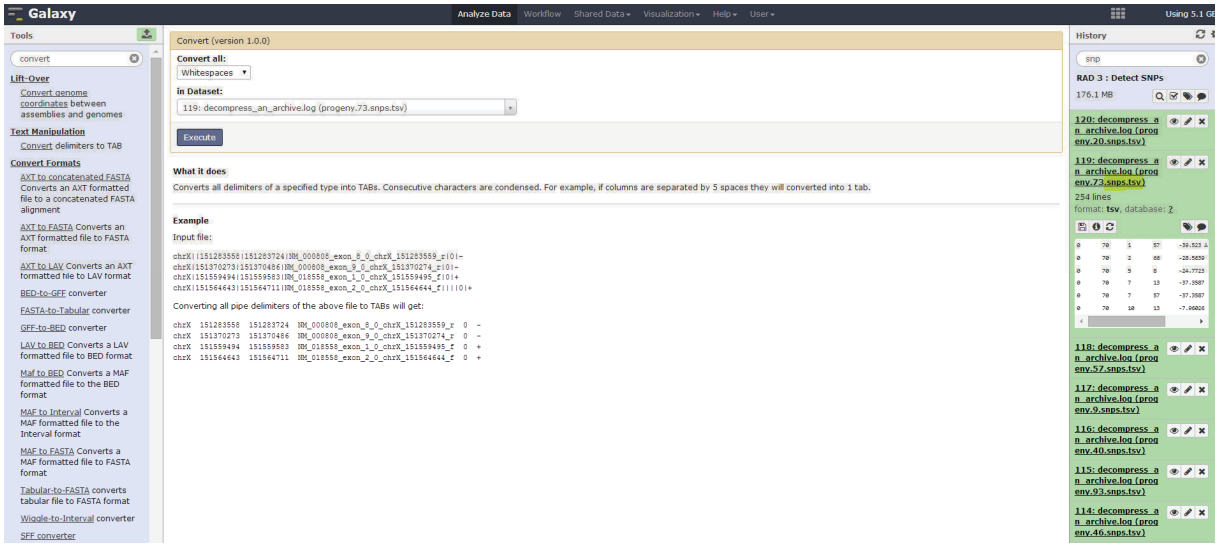
Extraire l'archive *Parents.zip* (outil ***Decompress an archive***) et ne pas oublier de rafraîchir l'historique une fois le job terminé. Modifier, si besoin, le type des fichiers extraits en fastq.

Relancer ***STACKS denovo*** en spécifiant l'usage "*Population*". Au passage, vous pouvez voir que les fichiers nouvellement décompressés sont directement utilisables en entrée de l'outil ***STACKS denovo.***

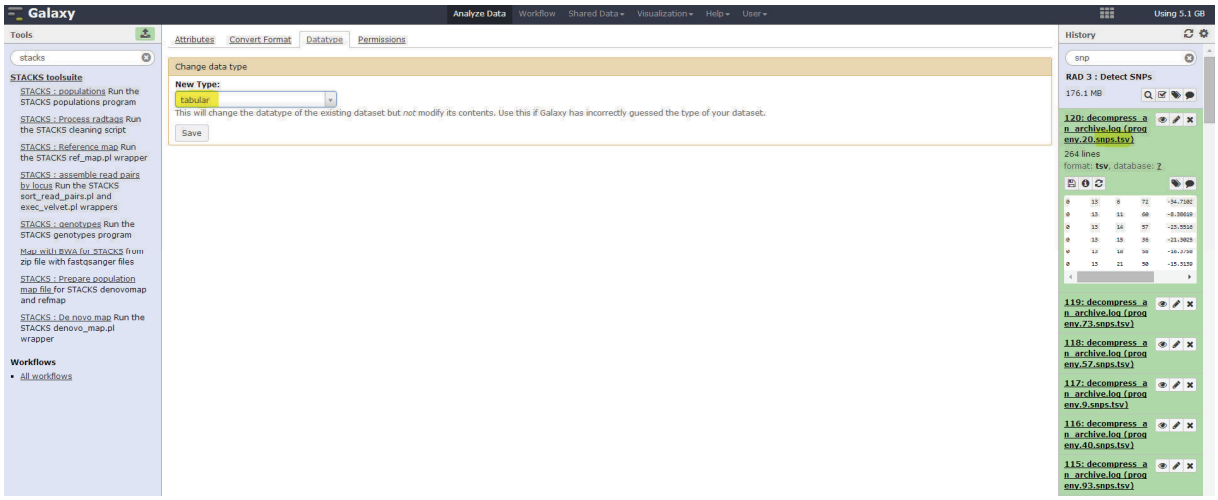
Que pouvez-vous dire des fichiers snps générés par les 2 usages? Vous serez probablement amenés à utiliser l'outil "compare two datasets"

Sélectionnez les snps identifiés chez l'individu femelle présentant un score (likelihood ratio) meilleur que le ¾ des scores. Vous serez probablement amenés à utiliser les outils "summary statistics" et "filter"

Si besoin (normalement pas dans le cas de figure présent), convertir les espaces en tabulations pour un des fichiers *.snps.tsv* via l'outil ***Convert***

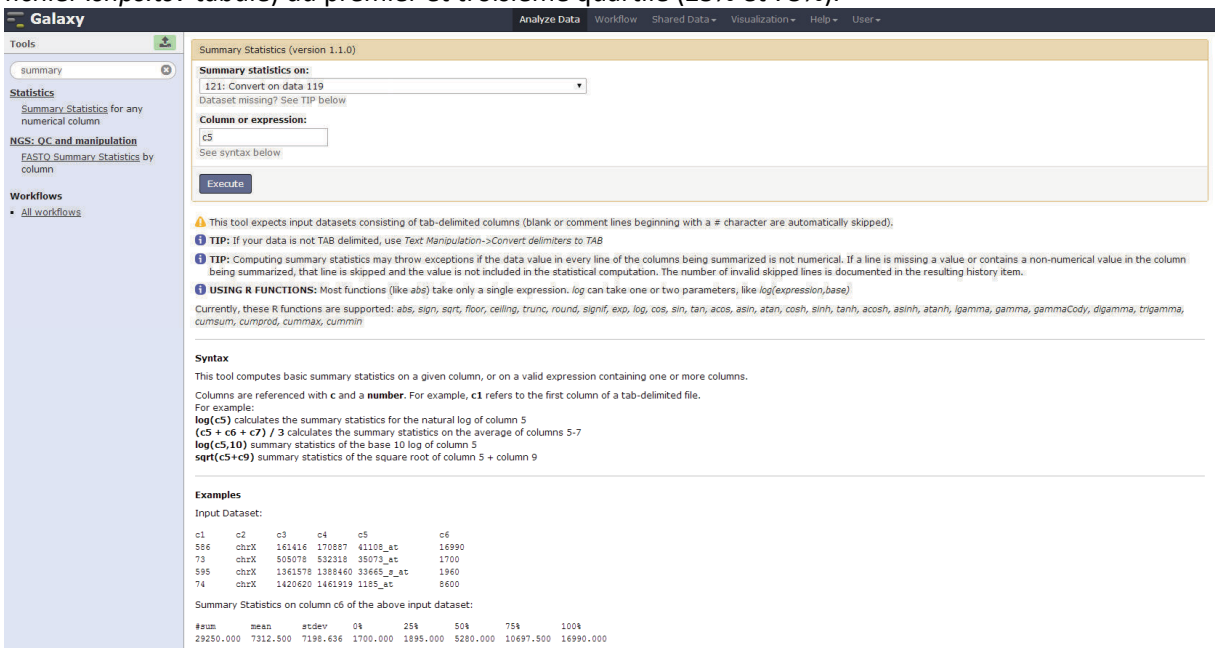


Il est aussi possible d'utiliser comme solution alternative, la modification directe du datatype à l'aide du petit crayon du dataset concerné.

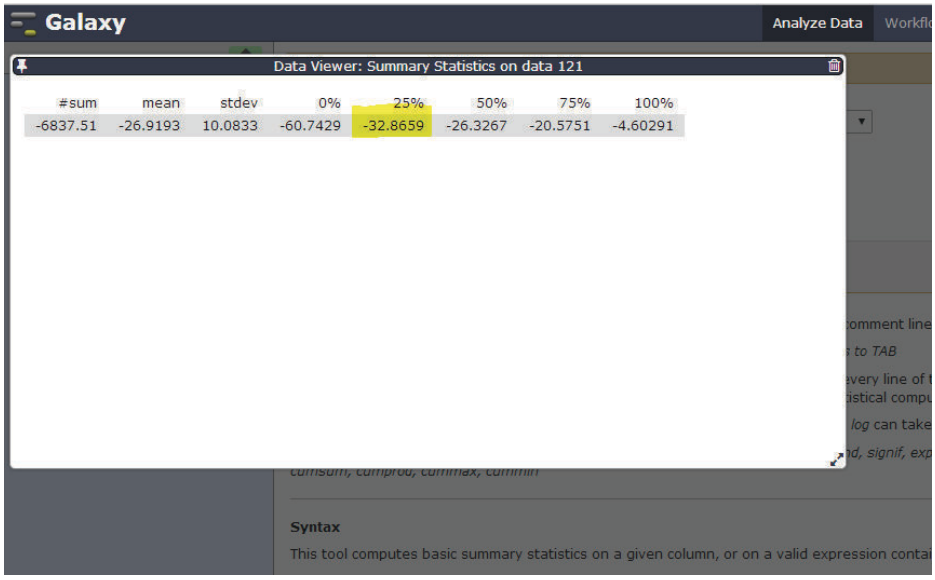


Filtrer le fichier tabulé obtenu par la valeur du *Likelihood ratio.* Plus ce score est important, plus il est significatif :

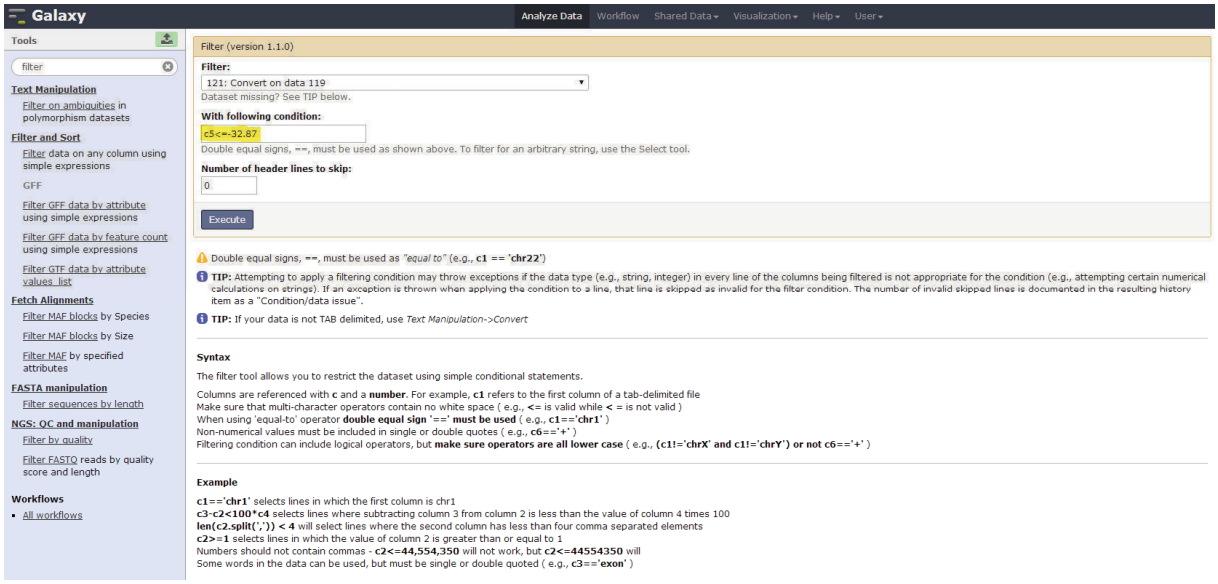
A l'aide de l'outil ***Summary Statistics***, récupérer les valeurs de likelihood ratio (5^{ième} colonne du fichier *.snps.tsv* tabulé) du premier et troisième quartile (25% et 75%).



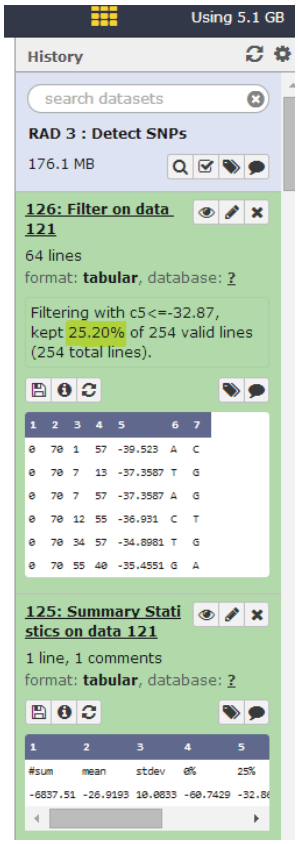
Ici, les valeurs sont toujours négatives, et celle du premier quartile, pour l'individu femelle, est de -32.8659 (-81.1918 avec la version 1.18 de Stacks ;))



Nous filtrerons le fichier en récupérant uniquement les SNP pour lesquels le Likelihood ratio est inférieur à cette valeur (ici -32.87).



Nous avons bien gardé ~25% des valeurs


Using 5.1 GB
History
search datasets
RAD 3 : Detect SNPs
176.1 MB
126: Filter on data 121
64 lines
format: tabular, database: ?
Filtering with c5<=-32.87, kept 25.20% of 254 valid lines (254 total lines).
1 2 3 4 5 6 7
0 70 1 57 -39.523 A C
0 70 7 13 -37.3587 T G
0 70 7 57 -37.3587 A G
0 70 12 55 -36.931 C T
0 70 34 57 -34.8981 T G
0 70 55 40 -35.4551 G A
125: Summary Statistics on data 121
1 line, 1 comments
format: tabular, database: ?
1 2 3 4 5
#sum mean stdev 0% 25%
-6837.51 -26.9193 10.0833 -60.7429 -32.8

II. Analyse RAD-seq sous Galaxy : La cartographie génétique

Cet exercice utilise des données générées par les développeurs de STACKS. Ils ont développé une carte génétique pour le Lépisosté tacheté (*Lepisosteus oculatus*) (poisson crocodile parfois surnommé brochet crocodile) et présentent ici les données d'un seul groupe de liaison. La carte génétique provient d'un croisement de type pseudo-test cross F1 entre 2 parents et 93 de leur descendants F1. Sont conservés pour l'exercice, uniquement des marqueurs apparaissant dans un unique groupe de liaison, et les lectures brutes des "stacks" qui ont contribué à ce groupe de liaison. Les fichiers sont déjà nettoyés, il ne sera donc pas nécessaire d'utiliser l'outil ***"STACKS : Process radtags Run the STACKS cleaning script"*** ici.

Les 95 fichiers fasta de départ sont donc déjà démultiplexés et se présentent comme suit :

```
>TTAAT_1_0046_17989_1193_1[67302]
TGCAGGCGAGGAAGTCACAGAGATCCCTGGCCAACTACTGTAGTTCGAACAGGAACCGAGCTGACAGGGCGCAGA
>TTAAT_1_0092_18487_11460_1[67302]
TGCAGGCGAGGAAGTCACAGAGATCCCTGGCCAACTACTGTAGTTCGAACAGGAACCGAGCTGACAGGGCGCAGA
>TTAAT_1_0094_8623_4235_1[67302]
TGCAGGCGAGGAAGTCACAGAGATCCCTGGCCAACTACTGTAGTTCGAACAGGAACCGAGCTGACAGGGCGCAGA
>TTAAT_1_0102_18666_9095_1[67302]
TGCAGGCGAGGAAGTCACAGAGATCCCTGGCCAACTACTGTAGTTCGAACAGGAACCGAGCTGACAGGGCGCAGA
>TTAAT_1_0114_6838_19507_1[67302]
TGCAGGCGAGGAAGTCACAGAGATCCCTGGCCAACTACTGTAGTTCGAACAGGAACTGAGCTGACAGGGCGCAGA
>TTAAT_1_0117_2046_2348_1[67302]
TGCAGGCGAGGAAGTCACAGAGATCCCTGGCCAACTACTGTAGTTCGAACAGGAACTGAGCTGACAGGGCGCAGA
>TTAAT_1_0038_16287_13120_1[22594]
TGCAGGCCTTGTGAAACTGAACAACACAAAAGGTTCTATCAATTAAAACCGCAGATAATTAGTTGTGTTTCTCCA
>TTAAT_1_0052_11753_10140_1[22594]
TGCAGGCCTTGTGAAACTGAACAACACAAAAGGTTCTATCAATTAAAACCGCAGATAATTAGTTGTGTTTCTCCA
>TTAAT_1_0054_1743_17715_1[22594]
TGCAGGCCTTGTGAAACTGAACAACACAAAAGGTTCTATCAATTAAAACCGCAGATAATTAGTTGTGTTTCTCCA
>TTAAT_1_0074_2389_17780_1[22594]
TGCAGGCCTTGTGAAACTGAACAACACAAAAGGTTCTATCAATTAAAACCGCAGATAATTAGTTGTGTTTCTCCA
>TTAAT_1_0087_18378_6512_1[22594]
```

Pour chaque séquence, on retrouve le barcode (ici TTAAT), et d'autres informations.

Créer un nouvel historique et renommer le (ex : Tuto GenOuest 1 RAD seq : Genetic map).

Comme les données sont les mêmes qu'utilisées dans la partie précédente, vous pouvez les rapatrier en utilisant l'option "*copy datasets*" de l'historique (URL d'origine : http://creskolab.uoregon.edu/stacks/tut_gar.php, les données : http://creskolab.uoregon.edu/stacks/tutorial/stacks_samples.tar.gz). Vous récupérer ainsi 2 archives :

-une regroupant les 2 jeux de données de séquences des parents : Parents.zip

-une regroupant les 93 jeux de données de séquences des descendants F1 : Progeny.zip

Nous sommes maintenant près à exécuter le pipeline STACKS denovo.

Sélectionner l'outil "***STACKS : De novo map Run the STACKS denovo_map.pl wrapper***"

Vous pouvez préciser certains paramètres. Ainsi, en sélectionnant le mode Advanced pour le paramètre ***Stack assembly options***, vous pouvez préciser:

-la profondeur minimum de la pile (-m), ici "Minimum number of identical". Ce paramètre, passé à ***ustacks***, contrôle le nombre de lectures correspondant exactement devant être trouvé pour créer une pile chez un individu. Nous pouvons sélectionner 3, une pile ne sera générée chez un individu que si au moins 3 lectures correspondent exactement.

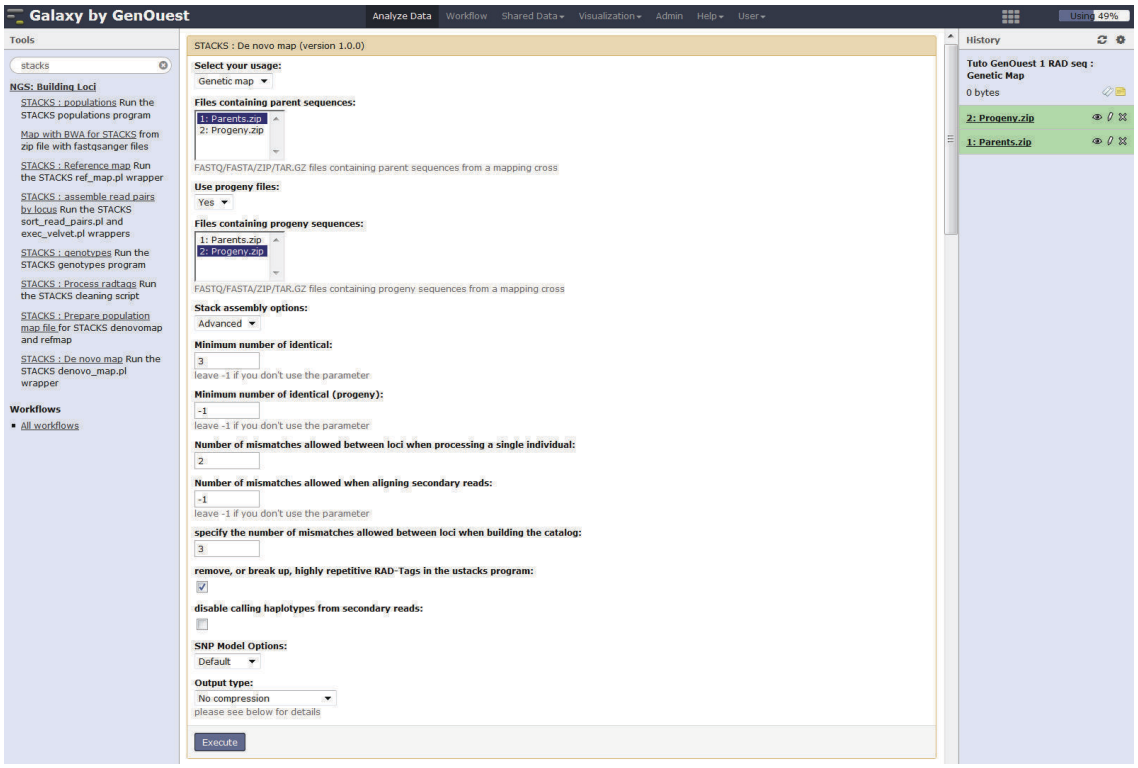
-la profondeur minimum de la pile (-P), ici "Minimum number of identical (progeny)". Ce paramètre, passé à ***ustacks***, contrôle le nombre de lectures correspondant exactement devant être trouvé pour créer une pile chez un descendant. Nous pouvons sélectionner 3, une pile ne sera générée chez un descendant que si au moins 3 lectures correspondent exactement.

-la distance maximum permise entre des piles pour qu'elles soient fusionnées en un locus potentiel chez un individu. Il s'agit du paramètre –M, ici "Number of mismatches allowed between loci when processing a single individual". Nous préciserons ici une distance maximale de 2 nucléotides.

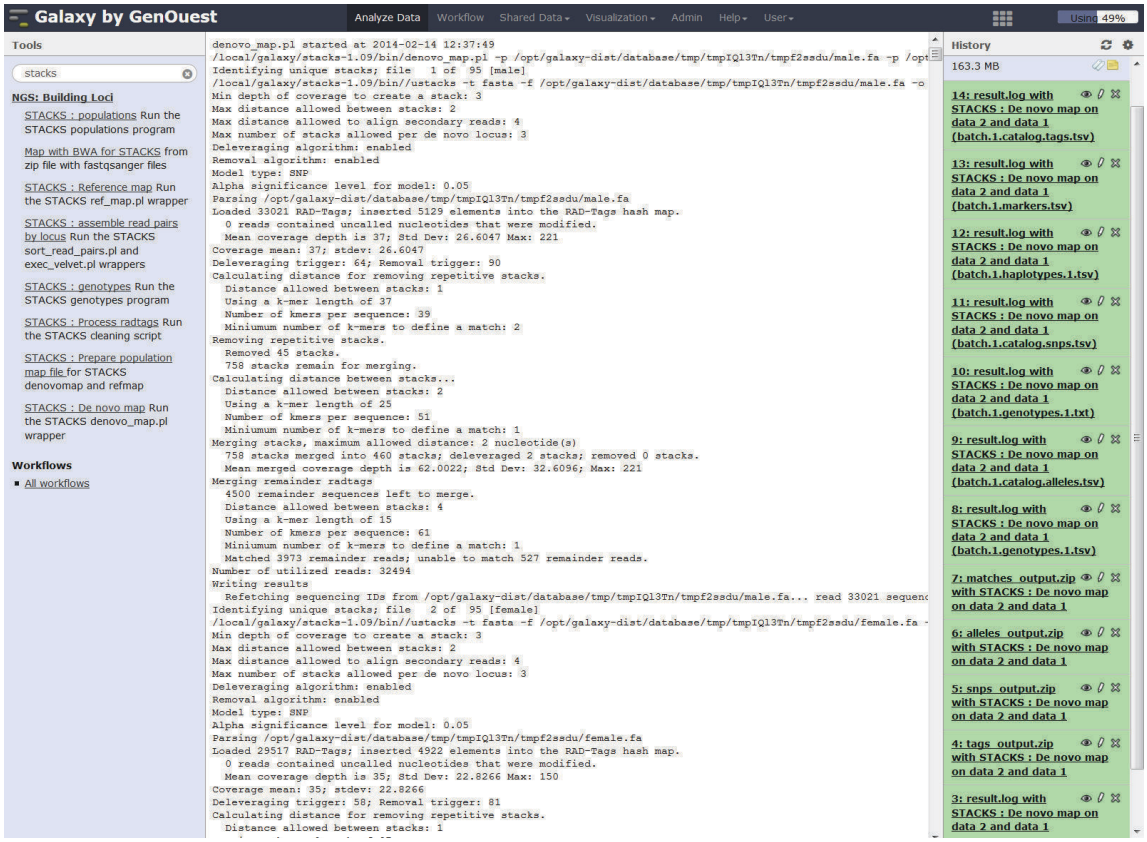
-le nombre maximal de différence entre tags. Ce paramètre permet notamment de pouvoir créer un locus de type homozygote dans le catalogue alors qu'il est en réalité hétérozygote. Ceci est pratique pour conserver les loci hétérozygotes entre parents mais homozygote.chez chacun d'entre eux. Il s'agit du paramètre –n, "specify the number of mismatches allowed between loci when building the catalog" ici. Nous pouvons fixer ce paramètre à 3.

-supprimer ou casser les RAD-tags très répétitifs. Il s'agit de supprimer les piles "bûcheronnes" de l'analyse et briser les piles modérément surdimensionnées. Nous cocherons ici cette option, correspondant au paramètre -t.

On choisira enfin le type de sortie : "compressed by categories"



Dans le cas présent, 12 jeux de données sont générés dans l'historique :



1. Un premier jeu de données nommé result.log, permettant de vérifier le bon déroulement du job.

```
denovo_map.pl started at 2014-02-14 12:37:49
/local/galaxy/stacks-1.09/bin/denovo_map.pl -p /
```

Comme vu lors de la première partie, nous pouvons voir le script lancé (ici denovo_map.pl), ainsi que la date et l'heure de départ du job. Ensuite, vient la ligne de commande lancée, `/local/galaxy/stacks-1.09/bin/denovo_map.pl -p…`
S'en suit le listing des étapes effectuées fichier par fichier :

```
Identifying unique stacks; file   1 of  95 [male]
/local/galaxy/stacks-1.09/bin//ustacks -t fasta -f /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpf2ssdu/male.fa -o /opt/ga
Min depth of coverage to create a stack: 3
Max distance allowed between stacks: 2
Max distance allowed to align secondary reads: 4
Max number of stacks allowed per de novo locus: 3
Deleveraging algorithm: enabled
Removal algorithm: enabled
Model type: SNP
Alpha significance level for model: 0.05
Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpf2ssdu/male.fa
Loaded 33021 RAD-Tags; inserted 5129 elements into the RAD-Tags hash map.
  0 reads contained uncalled nucleotides that were modified.
  Mean coverage depth is 37; Std Dev: 26.6047 Max: 221
Coverage mean: 37; stdev: 26.6047
Deleveraging trigger: 64; Removal trigger: 90
Calculating distance for removing repetitive stacks.
  Distance allowed between stacks: 1
  Using a k-mer length of 37
  Number of kmers per sequence: 39
  Miniumum number of k-mers to define a match: 2
Removing repetitive stacks.
  Removed 45 stacks.
  758 stacks remain for merging.
Calculating distance between stacks...
  Distance allowed between stacks: 2
  Using a k-mer length of 25
  Number of kmers per sequence: 51
  Miniumum number of k-mers to define a match: 1
Merging stacks, maximum allowed distance: 2 nucleotide(s)
  758 stacks merged into 460 stacks; deleveraged 2 stacks; removed 0 stacks.
  Mean merged coverage depth is 62.0022; Std Dev: 32.6096; Max: 221
Merging remainder radtags
  4500 remainder sequences left to merge.
  Distance allowed between stacks: 4
  Using a k-mer length of 15
  Number of kmers per sequence: 61
  Miniumum number of k-mers to define a match: 1
  Matched 3973 remainder reads; unable to match 527 remainder reads.
Number of utilized reads: 32494
Writing results
  Refetching sequencing IDs from /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpf2ssdu/male.fa... read 33021 sequence IDs.
```

Nous obtenons ainsi pour le premier des 95 fichiers, le parent male :

-la ligne de commande correspondant au lancement de ***ustacks***

-les paramètres renseignés lors de la soumission du job (Min depth of coverage, ….)

-le nombre total de lectures, qualifiées de RAD-tags, chargées (ici 33021) ainsi que le nombre d'éléments insérés dans la carte de hachage de RAD-tags (ici 5129).

-le nombre de lectures avec des nucléotides modifiés

-la profondeur de couverture moyenne

-le nombre de stacks répétés supprimés (ici 45)

-le nombre de stacks fusionnés car proches (ici 758 fusionnés en 460).

-le nombre de lectures restantes concordant avec les 460 stacks préalablement générés. Ici 3973.

Sur les 33021 lectures d'origines "RAD tagguées", seules 527 n'ont pas été utilisées. Il y a donc 32494 lectures finalement utilisées pour générer 460 stacks.

Ensuite est exécuté ***cstacks***.

```
/local/galaxy/stacks-1.09/bin//cstacks -b 1 -o /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9 -s
Number of mismatches allowed between stacks: 3
Loci matched based on sequence identity.
Constructing catalog from 2 samples.
Initializing new catalog...
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/male.tags.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/male.snps.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/male.alleles.tsv
Processing sample 2
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/female.tags.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/female.snps.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/female.alleles.tsv
Searching for sequence matches...
  Distance allowed between stacks: 3
  Using a k-mer length of 17
  Number of kmers per sequence: 59
  Miniumum number of k-mers to define a match: 8
  426 loci in the catalog, 29323 kmers in the catalog hash.
Merging matches into catalog...
  0 loci matched more than one catalog locus and were excluded.
Writing catalog...
```

Là encore, un rappel des paramètres utilisés est fait. Le catalogue de loci est ensuite créé à partir des échantillons parentaux. Ici, le catalogue créé contient 459 loci (voir *catalog.tags)* dont 426 provenant de l'"individu de référence" (male) et pouvant être partagé avec le second individu (female).

Enfin, ***sstacks*** est exécuté:

```
/local/galaxy/stacks-1.09/bin//sstacks -b 1 -c /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1 -s /
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.catalog.tags.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.catalog.snps.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.catalog.alleles.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/male.tags.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/male.snps.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/male.alleles.tsv
Searching for sequence matches...
426 stacks compared against the catalog containing 459 loci.
  426 matching loci, 0 contained no verified haplotypes.
  0 loci matched more than one catalog locus and were excluded.
  0 loci contained SNPs unaccounted for in the catalog and were excluded.
  695 total haplotypes examined from matching loci, 695 verified.
Outputing to file /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/male.matches.tsv
```

Pour chaque individu, ***sstacks*** détermine l'haplotype à chaque locus de chacun des individus du croisement. Il va alors comparer les stacks de chaque individu avec le catalogue de 459 loci. Ici, 695 haplotypes sont trouvés au total, tous vérifiés.

Enfin, ***genotypes*** est exécuté. Il récupère les loci contenant les marqueurs identifiés chez les parents, puis y associe les haplotypes des descendants. Si le premier parent présente les haplotypes *GA (ex : aatggtgt**G**gtccctcgt**A**c)* et *AC (ex : aatggtgt**A**gtccctcgt**C**c)*, et le second parent l'haplotype *GA (ex : aatggtgt**G**gtccctcgt**A**c)*, Stacks déclare un marqueur *ab*/*aa* pour ce locus. Le programme ***genotypes*** associe alors *GA* à *a*, et *AC* à *b* chez les parents puis scanne les descendants afin d'identifier quels haplotypes sont présents pour chacun d'entre eux et enregistrer les génotypes associés (soit *ab* ou *aa* dans le cas présent).

```
/local/galaxy/stacks-1.09/bin//genotypes -b 1 -P /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9 -t gen -r 1 -c -s  2>&1
Found 95 input file(s).
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.catalog.tags.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.catalog.snps.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.catalog.alleles.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/female.matches.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/male.matches.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_1.matches.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_10.matches.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_11.matches.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_12.matches.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_13.matches.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_14.matches.tsv
```

```
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_93.tags.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_93.snps.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_93.alleles.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_94.tags.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_94.snps.tsv
  Parsing /opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/progeny_94.alleles.tsv
42036 potential genotypes in 452 markers, 38874 populated; 1814 corrected, 1216 converted to heterozygotes, 598 unsupported homozygotes removed.
Writing 452 loci to genotype file, '/opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.genotypes_1.tsv'
Writing SQL markers file to '/opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.markers.tsv'
Writing SQL genotypes file to '/opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.genotypes_1.txt'
Writing 459 loci to observed haplotype file, '/opt/galaxy-dist/database/tmp/tmpIQl3Tn/tmpubNNI9/batch_1.haplotypes_1.tsv'
denovo_map.pl completed at 2014-02-14 12:38:16
```

Au final, 452 loci, marqueurs, sont conservés pour créer le fichier génotype ***batch_1.genotypes_1.tsv***. 459 loci sont répertoriés dans le fichier haplotype observé ***batch_1.haplotypes_1.tsv***. denovo_map.pl s'est exécuté en moins d'une minute. Les 42036 génotypes potentiels sont enregistrés dans le fichier ***batch_1.genotypes_1.txt***. 452 marqueurs sont enregistrés dans le fichier ***batch_1markers.tsv***…..

Réexécuter ***denovo_map*** en spécifiant l'option "*compressed all outputs*". Nous obtenons cette fois deux jeux de données en sortie, un fichier de log identique à celui généré lors de la précédente exécution de ***denovo_map*** et une archive nommée total_output.zip contenant tous les jeux de données de sortie de ***denov_map***. C'est cette archive qui sera utilisée par la suite comme fichier d'entrée de l'outil "***STACKS : genotypes Run the STACKS genotypes program***".

Exécuter l'outil "***STACKS : genotypes Run the STACKS genotypes program***" sur le jeu de données total_output.zip sans spécifier d'option. Cette nouvelle étape permet :

1/de spécifier un type particuliers de carte (ex: F1, F2, Doubles Haploid, Back Cross, Cross Pollination)

2/d'exporter les informations génotypiques dans différents formats spécifiques de programmes tiers de cartographie (ex: JoinMap, R/qtl). Il faut savoir que le format R/QTL pour un protocole F2 peut servir en entrée d'outils comme MapMaker ou Carthagène

3/de spécifier un nombre minimum de descendants et/ou une couverture minimum dans la pile pour considérer un locus

4/d'effectuer des corrections automatiques. Concernant ce dernier point, il est effectivement possible de demander au programme ***genotypes*** d'effectuer des corrections automatiques pour certaines erreurs comme la vérification des tags homozygotes dans la descendance pour assurer qu'un SNP n'est pas présent. En effet, si le modèle de détection de SNP ne peut pas identifier un site comme hétérozygote ou homozygote, le site est provisoirement marqué comme homozygote pour faciliter la recherche, par ***sstacks***, de concordances avec le catalogue de loci. Si un second allèle identifié dans le catalogue (i.e., chez les parents) est présent chez un des descendant à une faible fréquence (<10 des lectures de la pile considérée), le programme ***genotypes*** corrige le génotype. De même, il supprimera un génotype identifié comme homozygote chez un individu particuliers si le génotype au locus est supporté par moins de 5 lectures. Les génotypes corrigés sont alors notés en majuscule. Il faut savoir que l'utilisation de l'interface web de Stacks permet de modifier manuellement les génotypes. Ceci est intéressant notamment lorsqu'un allèle est sous séquencé chez un individu. Il est alors impossible pour Stacks de le dissocier d'une erreur de séquençage, et ce génotype sera marqué comme homozygote. Si l'utilisateur sait que l'allèle alternatif existe (en observant les génotypes d'autres individus par exemple), il peut alors le corriger manuellement avant de réexécuter ***genotypes***.

Reexécuter l'outil "***STACKS : genotypes*** ***Run the STACKS genotypes program***" sur le jeu de données total_output.zip en cochant cette fois l'option correction automatique.

Comparer les fichiers batch.1.genotypes.1.tsv générés par les deux jobs.

1. Les fichiers genotypes.tsv

Une ligne par locus, une colonne par individu (aa, ab, AB si correction automatique, bb, bc, …) avec le genotype observé à chacun des loci.

# Catalog ID	Marker	Cnt	Seg Dist	progeny_1	progeny_10	progeny_11	progeny_12	progeny_13	progeny_14
1	ab/aa	95	1	ab	ab	aa	ab	ab	ab
2	ab/aa	95	1	ab	ab	ab	ab	aa	aa
3	aa/ab	94	1	-	aa	ab	aa	ab	aa
4	ab/ac	94	1	aa	ac	-	ac	ab	bc
5	aa/ab	94	1	ab	aa	ab	aa	ab	aa
6	aa/ab	93	1	-	aa	ab	aa	AB	aa
7	ab/aa	92	1	-	aa	ab	aa	aa	ab
8	ab/aa	95	1	aa	ab	aa	aa	aa	aa
9	aa/ab	90	1	-	aa	AB	aa	-	aa
10	aa/ab	95	1	AB	aa	ab	aa	aa	AB
11	ab/ac	95	1	bc	ab	ac	ab	ac	aa
12	ab/ac	94	1	ac	aa	ac	aa	ac	ab
13	ab/aa	95	1	aa	aa	aa	aa	aa	ab
14	aa/ab	94	1	-	aa	ab	aa	ab	aa
15	aa/ab	95	1	ab	aa	ab	aa	ab	aa
16	ab/aa	84	1	-	ab	ab	ab	ab	aa
17	aa/ab	95	1	ab	ab	aa	ab	aa	aa
18	ab/aa	95	1	ab	ab	ab	ab	ab	aa
19	aa/ab	95	1	ab	aa	ab	aa	ab	aa
20	ab/aa	93	1	AB	aa	ab	ab	aa	aa
21	ab/ac	94	1	ab	bc	aa	bc	bc	ab
22	aa/ab	95	1	aa	ab	aa	ab	aa	AB
23	ab/aa	95	1	aa	aa	ab	aa	ab	ab

2. Les fichiers genotypes.txt

Une ligne par individu et pour chaque individu, à chaque locus du catalogue, le génotype.

# SQL ID	Batch ID	Catalog Locus ID	Sample ID	Genotype
0	1	1	3	ab
0	1	1	4	ab
0	1	1	5	aa
0	1	1	6	ab
0	1	1	7	ab
0	1	1	8	ab
0	1	1	9	ab
0	1	1	10	aa
0	1	1	11	aa
0	1	1	12	aa
0	1	1	13	ab
0	1	1	14	ab
0	1	1	15	ab
0	1	1	16	aa
0	1	1	17	AB
0	1	1	18	aa
0	1	1	19	aa
0	1	1	20	ab
0	1	1	21	ab
0	1	1	22	ab
0	1	1	23	ab
0	1	1	24	ab
0	1	1	25	aa

3. Les fichiers haplotypes.tsv

Cnt	Seg Dist	female	male	progeny_1	progeny_10	progeny_11	progeny_12	progeny_13	progeny_14
95	0.5	A/G	A	A/G	A/G	A	A/G	A/G	A/G
95	0.5	A/T	T	A/T	A/T	A/T	A/T	T	T
94	0.5	T	C/T	-	T	C/T	T	C/T	T
95	0.1	AA/GA	GA/GG	GA	GA/GG	GA	GA/GG	AA/GA	AA/GG
94	0.5	C	A/C	A/C	C	A/C	C	A/C	C
93	0.5	A	A/T	-	A	A/T	A	T	A
92	0.05	A/G	A	-	A	A/G	A	A	A/G
95	0.5	A/T	T	T	A/T	T	T	T	T
90	0.0005	A	A/G	-	A	A	A	G	A
95	0.5	G	C/G	C	G	C/G	G	G	C
95	0.5	AC/GC	AC/GG	GC/GG	AC/GC	AC/GG	AC/GC	AC/GG	AC
94	0.5	AT/CA	AA/CA	AA/CA	CA	AA/CA	CA	AA/CA	AT/CA
95	0.5	A/G	A	A	A	A	A	A	A/G

A la fin de cette partie, nous obtenons des génotypages d'individus pouvant être utilisés dans certains logiciels permettant de créer des cartes génétiques comme Carthagène ou Mapmaker

III. Analyse RAD-seq sous Galaxy : Construction de mini contigs à partir de séquences pairées

Il est possible d'assembler des mini-contigs à partir de lectures pairées issues de RAD-seq. Disposant ainsi de plusieurs centaines de nucléotides génomiques situés à proximité de chaque marqueur, l'ancrage de ces marqueurs à des librairies d'EST, et donc leur connexion à des gènes codant pour des protéines chez d'autres organismes, est facilité.

L'outil "***STACKS : assemble read pairs by locus*** ***Run the STACKS sort_read_pairs.pl and exec_velvet.pl wrappers***" permet de rassembler les lectures pairées associées à chaque stack au sein d'un fichier fasta. Une fois un fichier fasta généré par locus du catalogue, Velvet est utilisé pour assembler les lectures de chaque fichier.

Comme le fichier fasta doit porter dans l'identifiant de la séquence l'identifiant du locus du catalogue, il est possible d'ajouter manuellement des séquences à un locus. Ainsi, en plus des mini-contigs associés automatiquement aux loci par l'outil "***STACKS : assemble read pairs by locus*** ***Run the STACKS sort_read_pairs.pl and exec_velvet.pl wrappers***", nous pouvons ajouter manuellement des séquences d'EST disponibles, ou construites de novo en utilisant des données de RNA-seq. Cette étape peut être réalisée après avoir associé des séquences au catalogue de loci en utilisant Blast ou Bowtie.

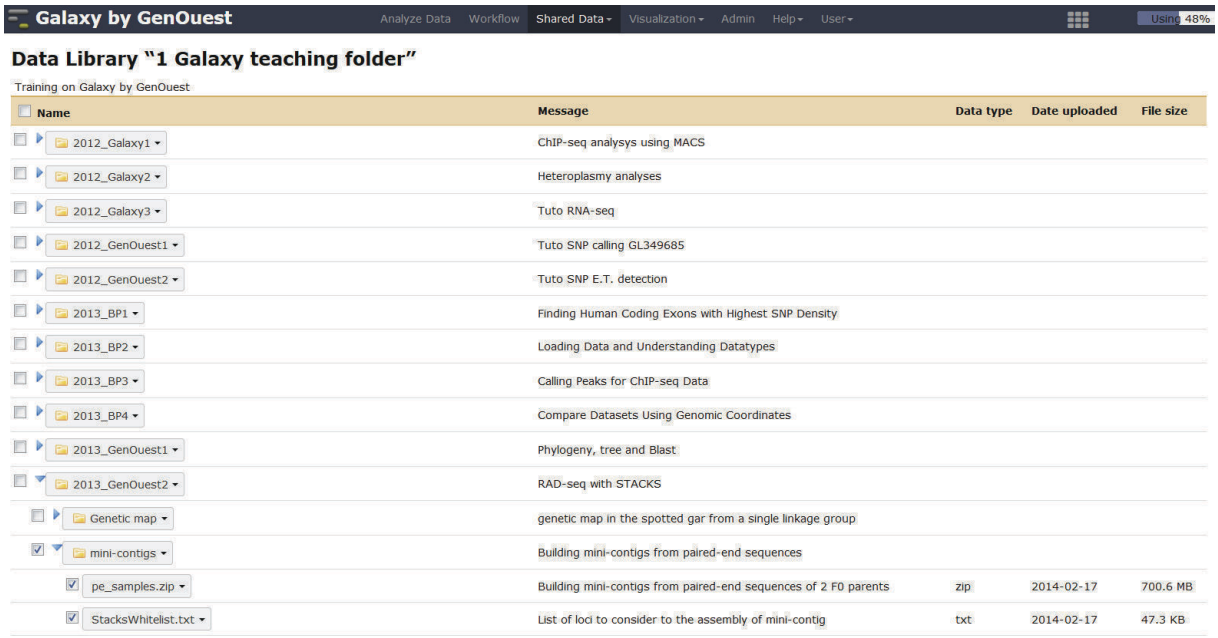
Créer un nouvel historique et renommer le (ex : Tuto GenOuest 2 RAD seq : Building mini-contig from paired-end sequences).

Récupération des données brutes dans Shared data/data libraries/1 Galaxy teaching folder/2013_GenOuest2/mini-contig (URL d'origine : http://creskolab.uoregon.edu/stacks/pe_tut.php) . Vous y trouverez

- une archive contenant 4 jeux de données de lectures pairées parentales : pe_samples.zip. Cette archive est constituée de 4 fichiers fastq, 2 par parent (***f0_female.1.fq***, ***f0_female.2.fq***, ***f0_male.1.fq*** et ***f0_male.2.fq***).

- un fichier texte "whitelist", reprenant les loci à considérer pour l'assemblage. Nous reviendrons sur ce point par la suite.

Sélectionner les 2 jeux de données, et cliquer sur le bouton "*GO*" après avoir vérifié que l'action sélectionnée était bien "*Import to current history*".



Vous débutez donc avec les deux jeux de données dans votre historique comme suit

Sélectionner l'outil "***STACKS : De novo map* Run the STACKS denovo_map.pl wrapper***".

Indiquez que vous souhaitez utiliser la fonction "Genetic map" et que vous n'utilisez pas de fichiers de descendants. ("Use progeny files": No).

STACKS ne gère pas les données pairées. En effet, si d'un côté du fragment le séquençage se fera toujours au niveau du même site (correspondant au site de coupure de l'enzyme), de l'autre, ce n'est pas le cas. Du coup, une belle pile présentant une forte couverture sera déterminée dans le premier sens de séquençage et pas dans l'autre. L'utilisation des lectures pairées entrainant de nombreux biais, en tout cas, présentant un intérêt très limité lors de l'exécution de "***STACKS : De novo map* Run the STACKS denovo_map.pl wrapper***", il est conseillé de n'utiliser que les lectures provenant d'un seul sens de séquençage. Toutefois, si vous disposez, comme c'est le cas ici, d'une archive comprenant des lectures pairées (avec toute les paires au complet), et que vous ne voulez pas créer

une archive particulière (avec uniquement les lectures provenant d'un sens de séquençage) pour "***STACKS : De novo map* Run the STACKS denovo_map.pl wrapper**" et une autre pour "***STACKS : assemble read pairs by locus* Run the STACKS sort_read_pairs.pl and exec_velvet.pl wrappers**", vous pouvez utiliser l'option "*Paired-end fastq files*". En cochant cette option lors de l'exécution de "***STACKS : De novo map* Run the STACKS denovo_map.pl wrapper**", vous indiquez à l'outil de trier les fichiers fastq présents dans l'archive, et de ne s'exécuter que sur les premiers jeux de données pairées, donc les données séquencées en forward. Ainsi, la même archive pourra être utiliser en entrée de l'outil "***STACKS : assemble read pairs by locus* Run the STACKS sort_read_pairs.pl and exec_velvet.pl wrappers**".

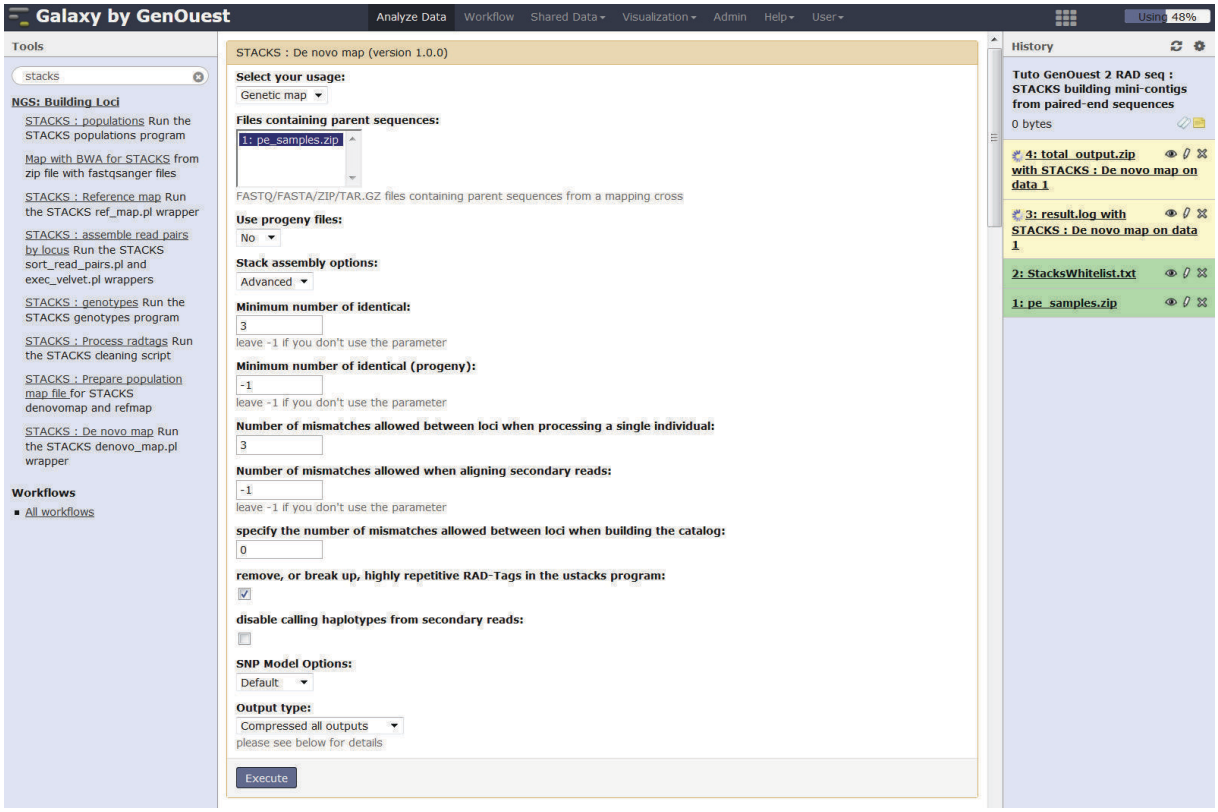
Vous pouvez préciser certains paramètres. Ainsi, en sélectionnant le mode Advanced pour le paramètre ***Stack assembly options***, vous pouvez préciser:

-la profondeur minimum de la pile (-m), ici "Minimum number of identical". Ce paramètre, passé à ustacks, contrôle le nombre de lectures correspondant exactement devant être trouvé pour créer une pile chez un individu. Nous pouvons sélectionner 3, une pile ne sera générée chez un individu que si au moins 3 lectures correspondent exactement.

-la distance maximum permise entre des piles pour qu'elles soient fusionnées en un locus potentiel chez un individu. Il s'agit du paramètre –M, ici "Number of mismatches allowed between loci when processing a single individual". Nous préciserons ici une distance maximale de 3 nucléotides.

-supprimer ou casser les RAD-tags très répétitifs. Il s'agit de supprimer les piles "bûcheronnes" de l'analyse et briser les piles modérément surdimensionnées. Nous cocherons ici cette option, correspondant au paramètre -t.

On choisira enfin le type de sortie : "compressed all outputs" pour générer une archive contenant tous les résultats du job.



Dans l'archive générée, nous trouvons plusieurs jeux de données:

- f0_male.2.snps.tsv : répertorie tous les snps identifiés chez l'individu considéré, ici 50 353 SNPs.

- f0_male.2.tags.tsv : présente le détail de tous les loci RAD identifiés chez l'individu considéré. Il montre la séquence consensus pour un locus et les piles de lectures constitutives.

- f0_male.2.alleles.tsv : répertorie les haplotypes identifiés à chaque locus.

- f0_male.2.matches.tsv : répertorie les haplotypes vérifiés et se référant aux loci du catalogue. Ici 292 499 haplotypes sont répertoriés. Il s'agit du dernier fichier généré.

Trois fichiers répertorient le même type d'information (loci, SNPs et allèles) mais à l'échelle de tous les échantillons.

- batch_1.catalog.tags.tsv : qui présente les 382 005 loci enregistrés dans le catalogue.

- batch_1.catalog.snps.tsv : qui présente les 72 466 SNPs enregistrés dans le catalogue.

- batch_1.catalog.alleles.tsv : qui présente les 146 753 allèles enregistrés dans le catalogue.

- batch_1.haplotypes_1.tsv : qui présente les 381 735 haplotypes enregistrés dans le catalogue

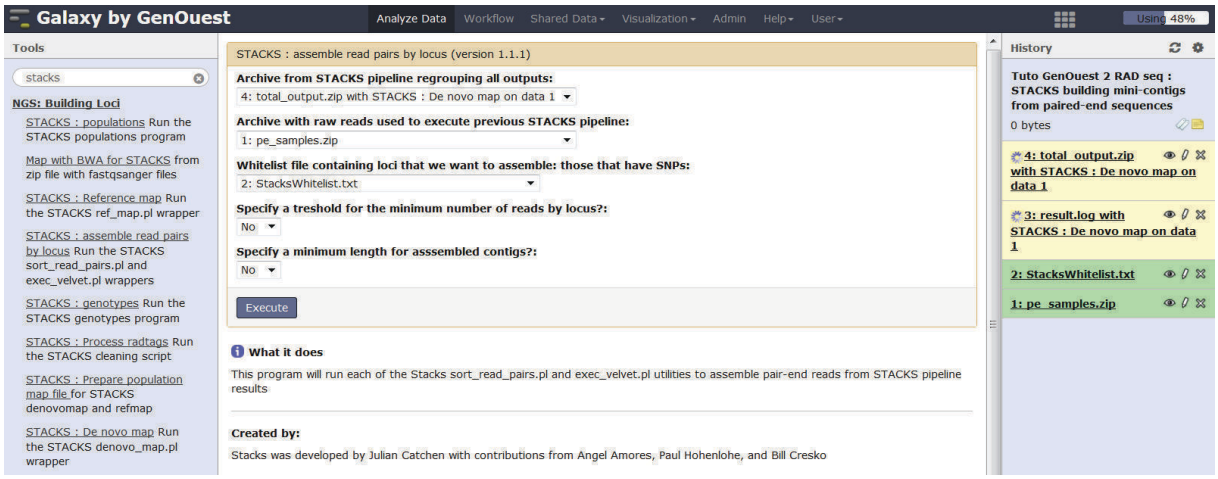
- batch_1.genotypes_1.tsv : vide car pas de descendants indiqués.

- batch_1.genotypes_1.txt : vide car pas de descendants indiqués.

- batch_1.markers.tsv : vide car pas de descendants indiqués.

En attendant que le job se termine (~50min), vous pouvez débuter la partie génomique des pop.

Sélectionner l'outil "***STACKS : assemble read pairs by locus*** ***Run the STACKS sort_read_pairs.pl and exec_velvet.pl wrappers***".



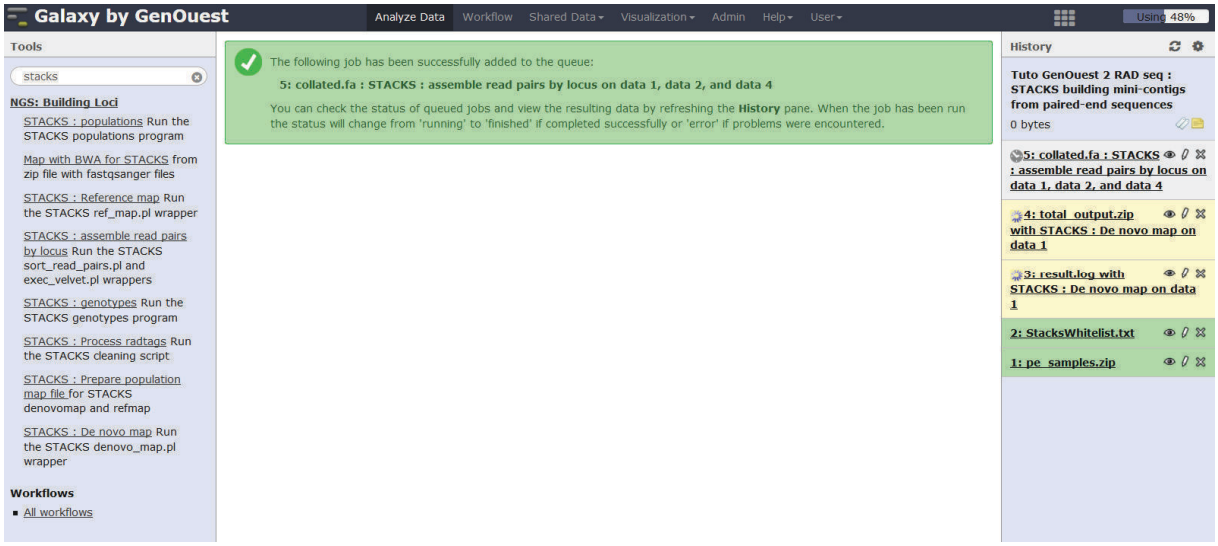
Indiquez :

- l'archive générée par l'outil "***STACKS : De novo map*** ***Run the STACKS denovo_map.pl wrapper***". Ici "***4: total_output.zip with STACKS : De novo map on data 1***".

- l'archive regroupant les 4 fichiers de lectures "***1: pe_samples.zip***".

- Si besoin, une liste des loci à assembler. Ici "***2: StacksWhitelist.txt***". Cette liste peut correspondre à l'ensemble des loci du catalogue de loci présentant des SNPs. Pour la créer, vous pouvez par exemple utiliser l'outil "***Cut*** ***columns from a table***" pour récupérer la troisième colonne du jeu de données "**batch_1.catalog.snps.tsv**" (qui répertorie tous les loci présentant des SNPs dans le catalogue de loci) généré par "***STACKS : De novo map*** ***Run the STACKS denovo_map.pl wrapper***", puis exécuter l'outil "***remove tab dupes*** ***Finds duplicates in tab format files***" sur la colonne coupée afin de disposer d'une whitelist sans redondance.

Nous ne spécifierons pas de nombre ou taille minimum de lectures pour assembler un locus.



IV. Analyse RAD-seq sous Galaxy : La génomique des populations

Nous allons travailler à partir des données de la publication d'*Hohenlohe et al. 2010*.

OPEN ACCESS Freely available online PLoS GENETICS

Population Genomics of Parallel Adaptation in Threespine Stickleback using Sequenced RAD Tags

Paul A. Hohenlohe[1☯], Susan Bassham[1☯], Paul D. Etter[2], Nicholas Stiffler[3], Eric A. Johnson[2], William A. Cresko[1*]

1 Center for Ecology and Evolutionary Biology, University of Oregon, Eugene, Oregon, United States of America, **2** Institute of Molecular Biology, University of Oregon, Eugene, Oregon, United States of America, **3** Genomics Core Facility, University of Oregon, Eugene, Oregon, United States of America

Abstract

Next-generation sequencing technology provides novel opportunities for gathering genome-scale sequence data in natural populations, laying the empirical foundation for the evolving field of population genomics. Here we conducted a genome scan of nucleotide diversity and differentiation in natural populations of threespine stickleback (*Gasterosteus aculeatus*). We used Illumina-sequenced RAD tags to identify and type over 45,000 single nucleotide polymorphisms (SNPs) in each of 100 individuals from two oceanic and three freshwater populations. Overall estimates of genetic diversity and differentiation among populations confirm the biogeographic hypothesis that large panmictic oceanic populations have repeatedly given rise to phenotypically divergent freshwater populations. Genomic regions exhibiting signatures of both balancing and divergent selection were remarkably consistent across multiple, independently derived populations, indicating that replicate parallel phenotypic evolution in stickleback may be occurring through extensive, parallel genetic evolution at a genome-wide scale. Some of these genomic regions co-localize with previously identified QTL for stickleback phenotypic variation identified using laboratory mapping crosses. In addition, we have identified several novel regions showing parallel differentiation across independent populations. Annotation of these regions revealed numerous genes that are candidates for stickleback phenotypic evolution and will form the basis of future genetic analyses in this and other organisms. This study represents the first high-density SNP–based genome scan of genetic diversity and differentiation for populations of threespine stickleback in the wild. These data illustrate the complementary nature of laboratory crosses and population genomic scans by confirming the adaptive significance of previously identified genomic regions, elucidating the particular evolutionary and demographic history of such regions in natural populations, and identifying new genomic regions and candidate genes of evolutionary significance.

Citation: Hohenlohe PA, Bassham S, Etter PD, Stiffler N, Johnson EA, et al. (2010) Population Genomics of Parallel Adaptation in Threespine Stickleback using Sequenced RAD Tags. PLoS Genet 6(2): e1000862. doi:10.1371/journal.pgen.1000862

Editor: David J. Begun, University of California Davis, United States of America

Received October 20, 2009; **Accepted** January 28, 2010; **Published** February 26, 2010

Funding: This work was funded by grants from the National Science Foundation (IOS-0642264) and from the National Institutes of Health (1R24GM079486-01A1 and Ruth L. Kirschstein National Research Service Award F32 GM078949). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: wcresko@uoregon.edu

☯ These authors contributed equally to this work.

Introduction

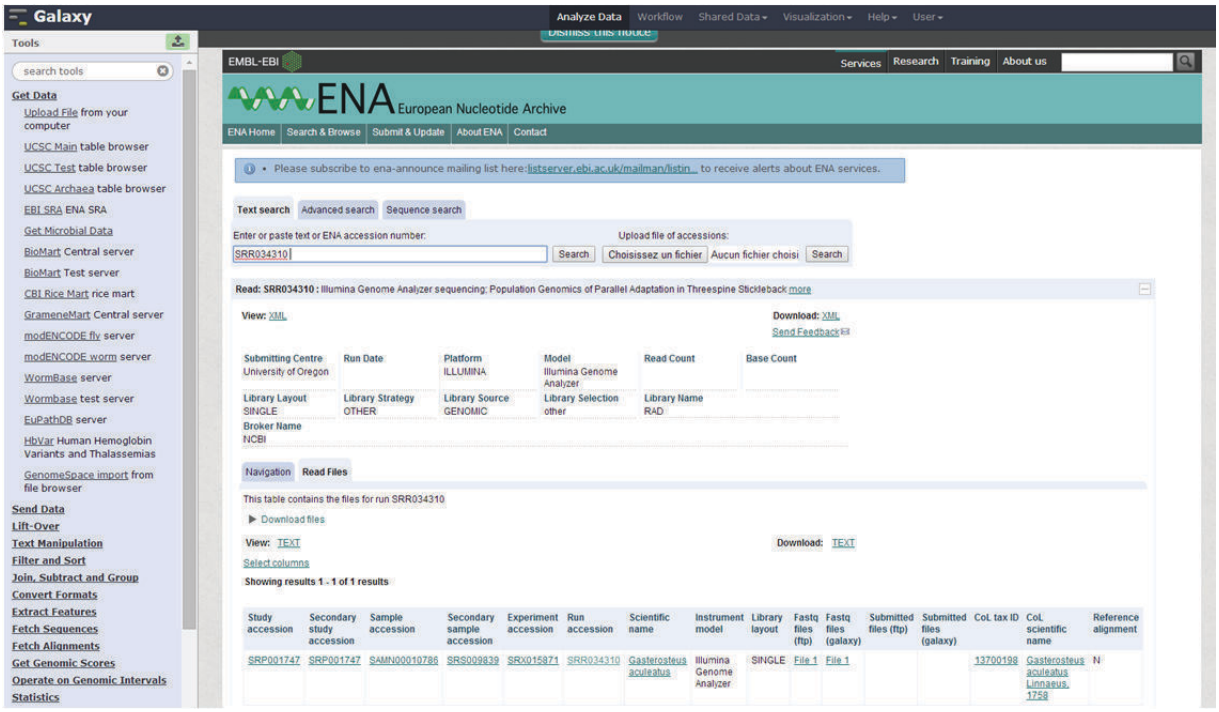
Population genetics provides a rich and mathematically rigorous framework for understanding evolutionary processes in natural populations. This theory was built over the last hundred years by modeling the processes of selection, genetic drift, mutation and migration in spatially distributed populations [1–6]. The field has concentrated primarily on the dynamics of one or a small number of genetic loci, largely because of methodological limitations. However, genes are not islands, but rather form part of a genomic community, integrated both by physical proximity on chromosomes and by various evolutionary processes [7–10]. With technological advances, such as Next Generation Sequencing (NGS) [11–13], the emerging field of population genomics now allows us to address evolutionary processes at a genomic scale in natural populations [14–20]. Population genetic measures like Wright's *F* statistics [2,21,22], traditionally viewed as point estimates, can now be examined as continuous distributions across a genome [23–29]. As a result, in addition to estimating genome-wide averages for such statistics, we can identify specific genomic regions that exhibit significantly increased or decreased differentiation among populations, indicating regions that have likely been under strong diversifying or stabilizing natural selection [9,30–41]. These signatures of selection can then be used to identify candidate pathways, genes and alleles for targeted functional analyses [42–47].

An excellent opportunity for this type of population genomics approach exists in the threespine stickleback, *Gasterosteus aculeatus* [48–50]. This small fish is distributed holarctically and inhabits a large number of marine, estuarine and freshwater habitats in Asia, Europe and North America. In many regions replicate extant freshwater stickleback populations have been independently derived from oceanic ancestors when stickleback became isolated postglacially in newly created freshwater habitats [49,51]. Population genetic data support this inference, and also indicate that present day oceanic populations can be used as surrogates for stock that gave rise to nearby derived freshwater populations [52–64]. Because of the varied selection regimes in novel habitats,

La génétique des populations est une très vieille discipline riche en théories mathématiques, et utilisant différentes approches statistiques permettant l'inférence de paramètres à partir de données génétiques. Ces statistiques se retrouvent à travers la détermination de la diversité nucléotidique (π), ou de coefficients de differentiation (i.e. Fst), ainsi que les mesures de covariances génétique comme le déséquilibre de liaison (D and D'). Cependant, à cause de limitations méthodologiques notamment, la majorité des travaux théoriques, statistiques et empiriques en génétique des populations s'est concentré sur un nombre restreint de loci. Avec l'avènement des NGS, des dizaines ou centaines de milliers de marqueurs génétiques peuvent désormais être examinés sur de nombreux individus, permettant à la discipline nommée génomique des population de devenir une réalité. Une nouvelle activité très excitante en génomique des populations réside dans le fait d'identifier des signatures de sélection dans les populations sauvages. Aujourd'hui, on travaillera sur des données de RAD à partir d'échantillonnage océanique et d'eau douce de populations d'épinoches.

Sous Galaxy, commencez par créer un nouvel historique (ex: "*RAD 1 : SNP calling*").

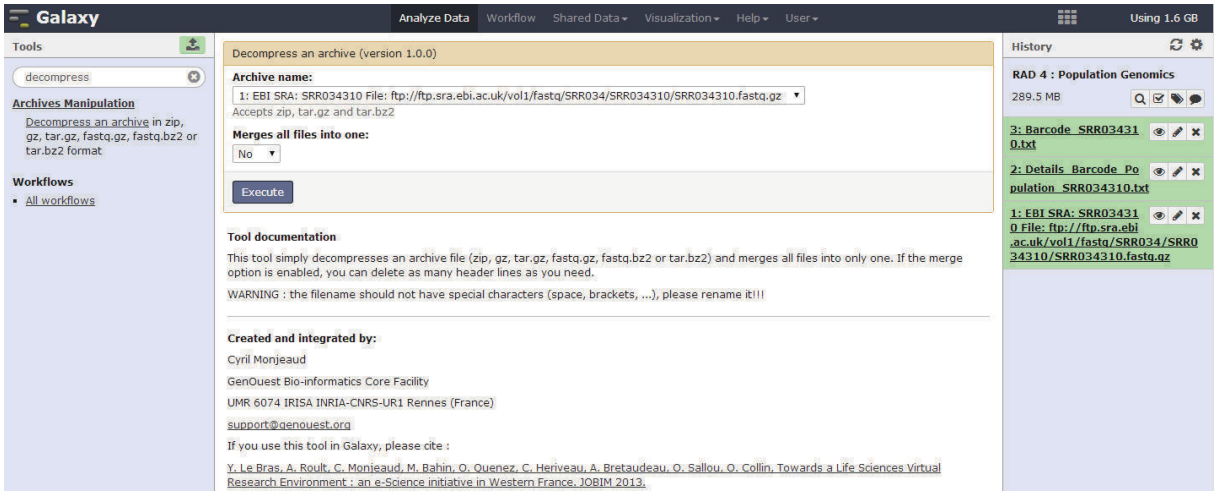
Les données seront récupérées via l'outil EBI SRA ENA SRA de la section Get Data. Les numéros d'accession des 6 jeux de données vont de SRR034310 à SRR034316 (attention, le moteur de recherche est sensible à la casse ;( ). Nous chargerons ici le jeux de données SRR034310 en cliquant sur (Fastq files (galaxy))

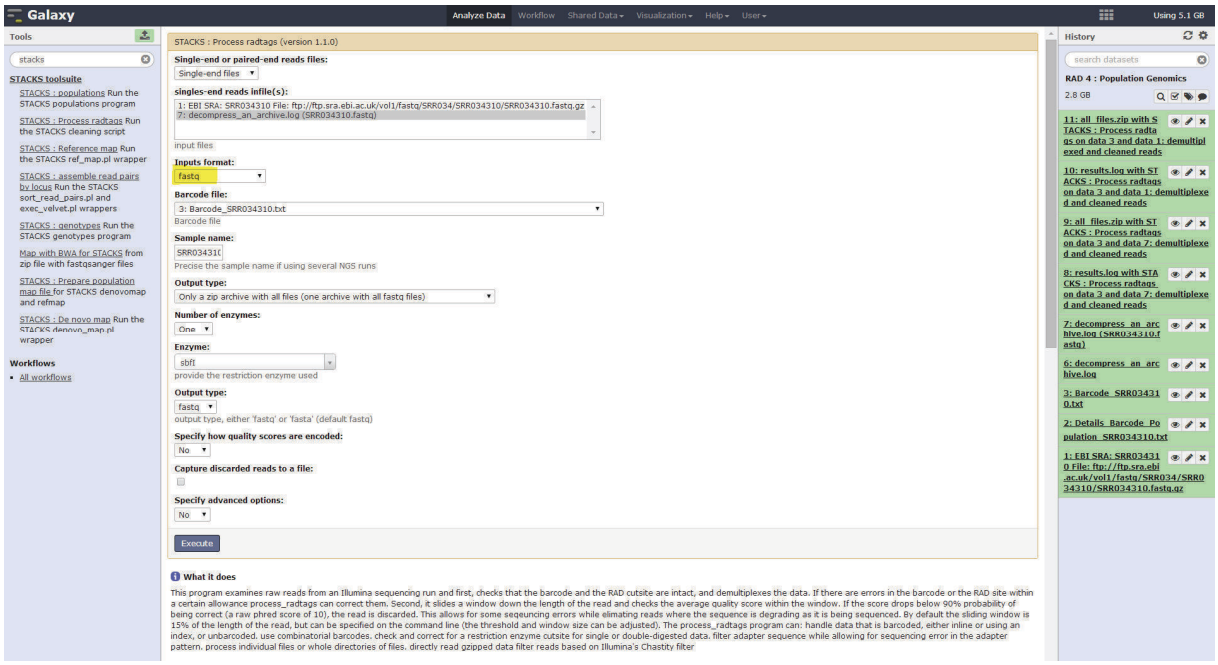


Récupération des informations concernant les barcodes et les populations liées à SRR034310 dans Shared data/data libraries/1 Galaxy teaching folder/2013_GenOuest2/population Genomics/Hohenlohe 2010

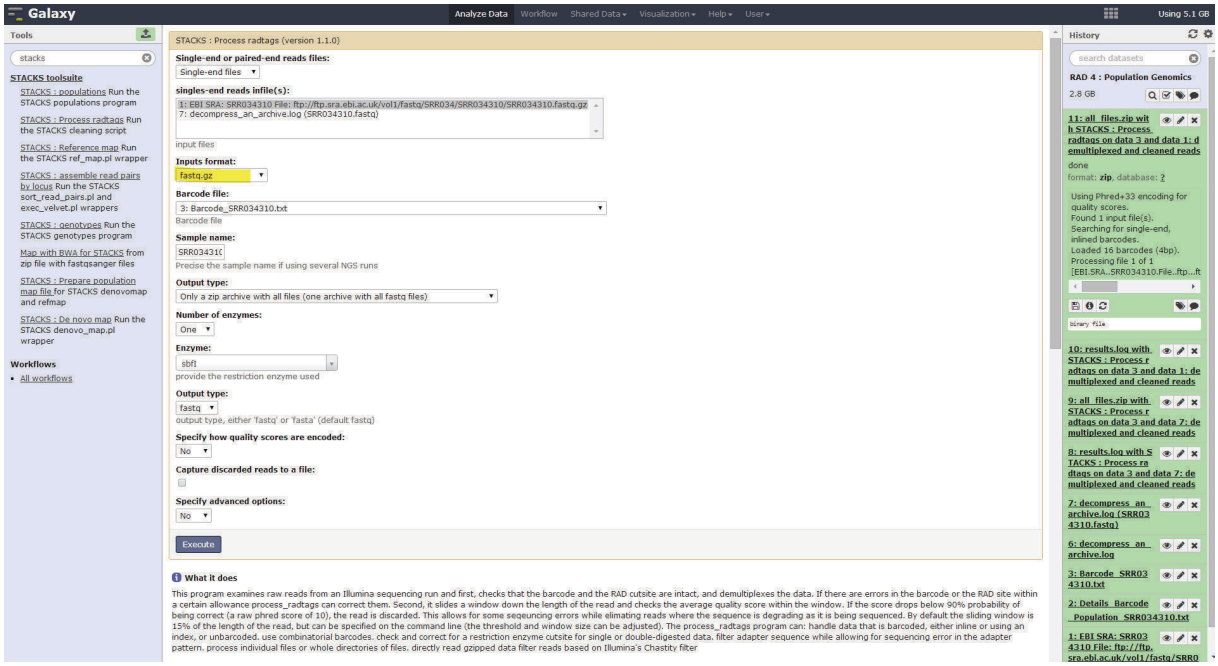
Deux possibilités ensuite.

- Si l'archive utilisée n'est pas au format *fastq.gz*, elle doit être décompressée avant utilisation dans ***Process_radtags***





- Si l'archive vient de SRA, elle doit probablement être au format *fastq.gz*. Dans ce cas, il suffit de spécifier le format comme indiqué ici :



L'outil STACKS : Process Radtags produit un fichier de log. Examinez le et répondez aux questions suivantes:

Combien de lectures brutes étaient présentes ?

Combien ont été retenues?

Sur celles non retenues, quelles étaient les raisons?

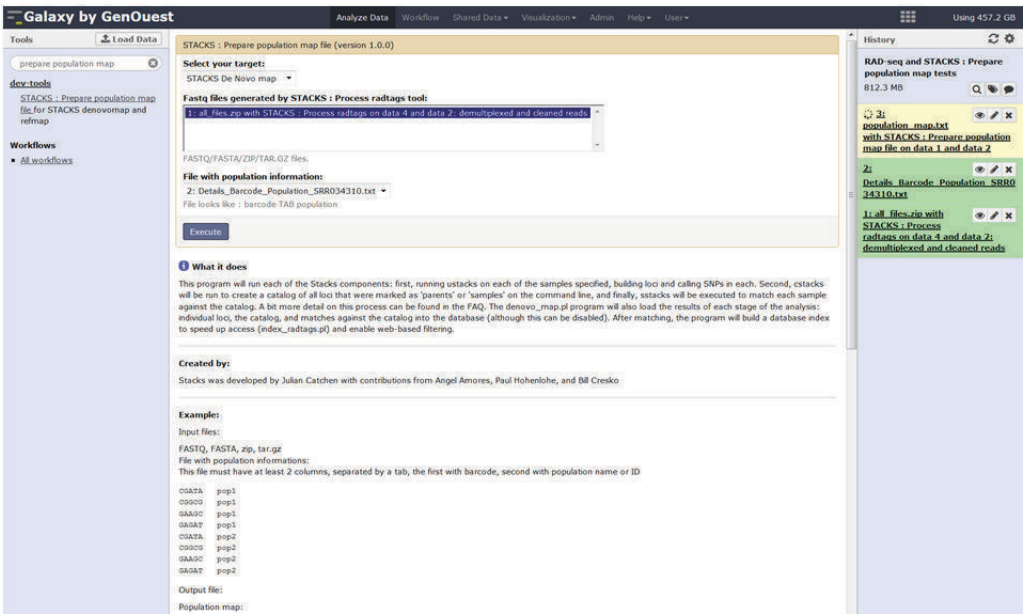
Que vous apprennent ce résultat au sujet de l'analyse de données et du design de barcodes en général?

Vous pourrez effectuer plusieurs exécution de ***STACKS : Process Radtags*** en jouant sur le score seuil affecté à une fenêtre glissante et en ajoutant les barcodes mentionnés dans le log précédent.

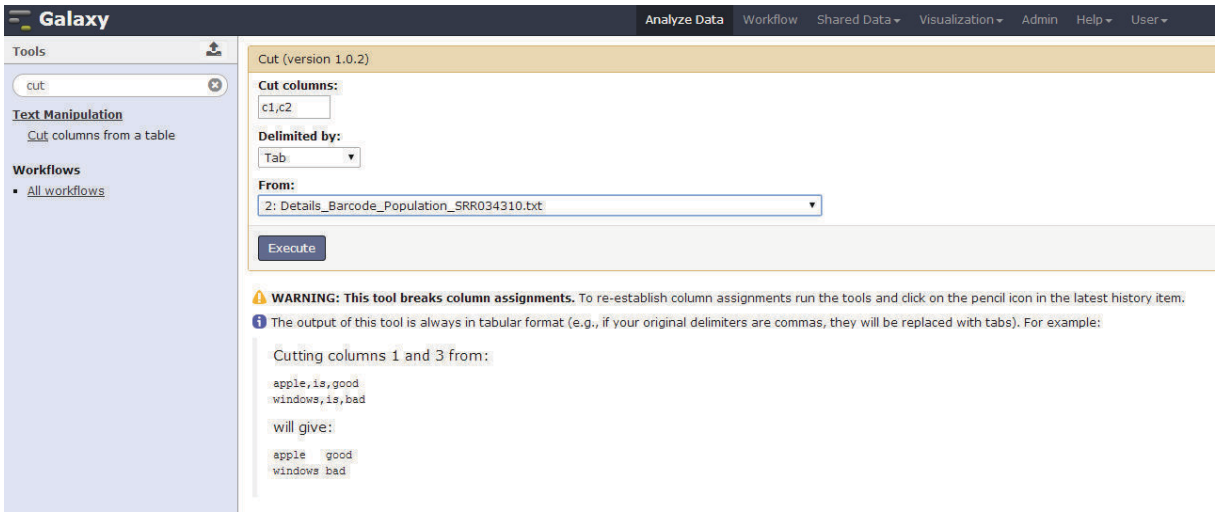
Quel est l'effet de l'augmentation du seuil du score?

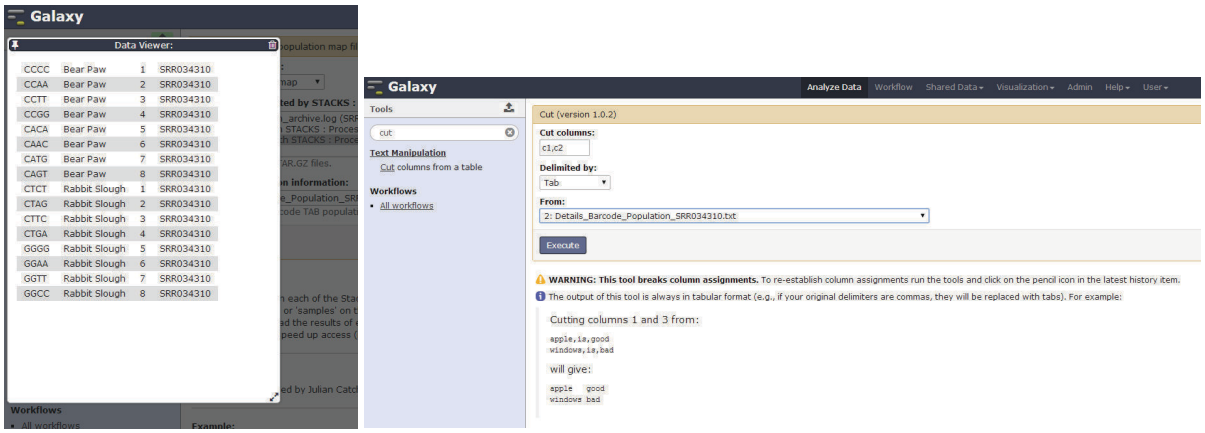
Vous pourrez également jouer sur l'ensemble des paramètres de ***STACKS : Process Radtags*** en spécifiant notamment une mauvaise enzyme de restriction, en faisant varier le score seuil de la fenêtre glissante

Afin de générer un fichier de type population map possédant et mappant les bonnes informations, nous proposons l'utilisation de l'outil ***STACKS : Prepare population map file***.

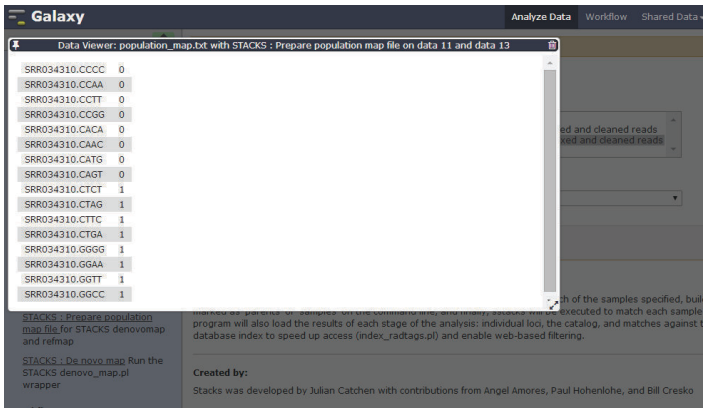


Il prend en entrée les fichiers fastq demultiplexés (organisés ou non dans une archive) ainsi qu'un fichier permettant de lier les barcodes aux populations, ici Details_Barcode_Population_SRR034310.txt présentant exactement 2 colonnes, la première colonne contenant les barcodes, la seconde le nom de la population. Pour faire ce fichier, il est possible de partir d'un jeu de donnée contenant plusieurs colonnes et d'utiliser l'outil Galaxy ***cut*** pour ne garder que les 2 colonnes nécessaires et placées dans le bon ordre (voir ci-dessous). Dans le cas présent, vérifier le format de votre fichier "Details…" pour savoir si cette manipulation est nécessaire.





Nous obtenons finalement un fichier de type Stacks "*population map*" :



Nous pouvons à présent exécuter ***STACKS : de novo map*** sur les individus étudiés. Il est possible de charger des fichiers FastQ ou directement l'archive "*all_files.zip*" générée par l'outil ***STACKS : Process Radtags*** comme c'est le cas ici. Il faudra alors préciser un fichier de type STACKS "*population map*" si l'analyse se fait sur plusieurs populations

Vous pouvez préciser certains paramètres. Ainsi, en sélectionnant le mode Advanced pour le paramètre ***Stack assembly options***, vous pouvez préciser:

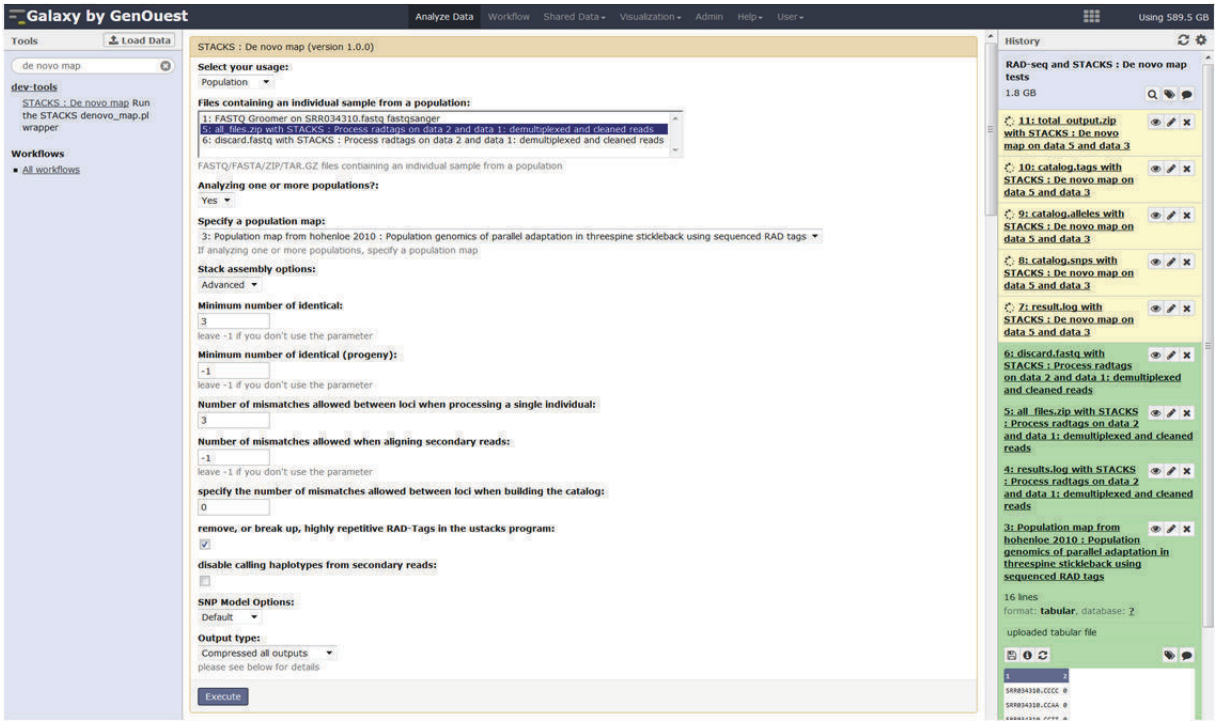
-la profondeur minimum de la pile (-m), ici "Minimum number of identical". Ce paramètre, passé à ***ustacks***, contrôle le nombre de lectures correspondant exactement devant être trouvé pour créer une pile chez un individu. Nous pouvons sélectionner 3, une pile ne sera générée chez un individu que si au moins 3 lectures correspondent exactement.

-la distance maximum permise entre des piles pour qu'elles soient fusionnées en un locus potentiel chez un individu. Il s'agit du paramètre –M, ici "Number of mismatches allowed between loci when processing a single individual". Nous préciserons ici une distance maximale de 2 nucléotides.

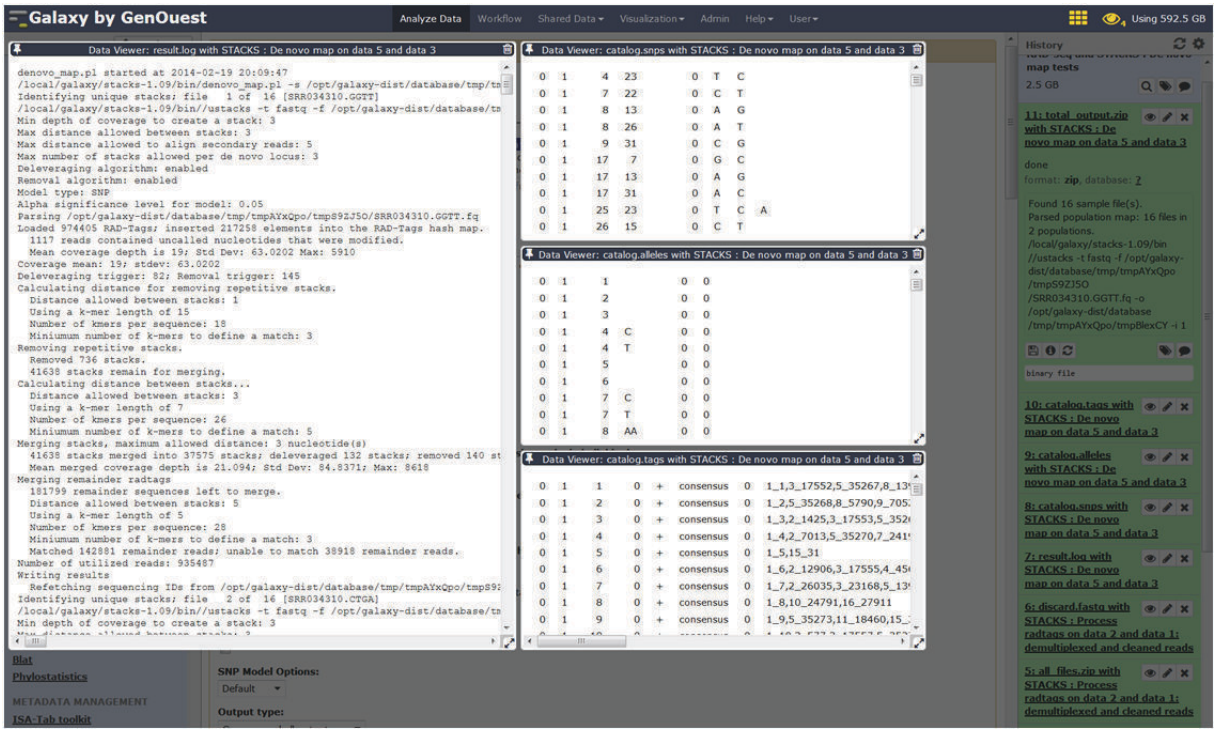
-le nombre maximal de différence entre tags. Ce paramètre permet notamment de pouvoir créer un locus de type homozygote dans le catalogue alors qu'il est en réalité hétérozygote. Ceci est pratique pour conserver les loci hétérozygotes entre parents mais homozygote.chez chacun d'entre eux. Il s'agit du paramètre –n, "specify the number of mismatches allowed between loci when building the catalog" ici. Nous pouvons fixer ce paramètre à 3.

-supprimer ou casser les RAD-tags très répétitifs. Il s'agit de supprimer les piles "bûcheronnes" de l'analyse et briser les piles modérément surdimensionnées. Nous cocherons ici cette option, correspondant au paramètre -t.

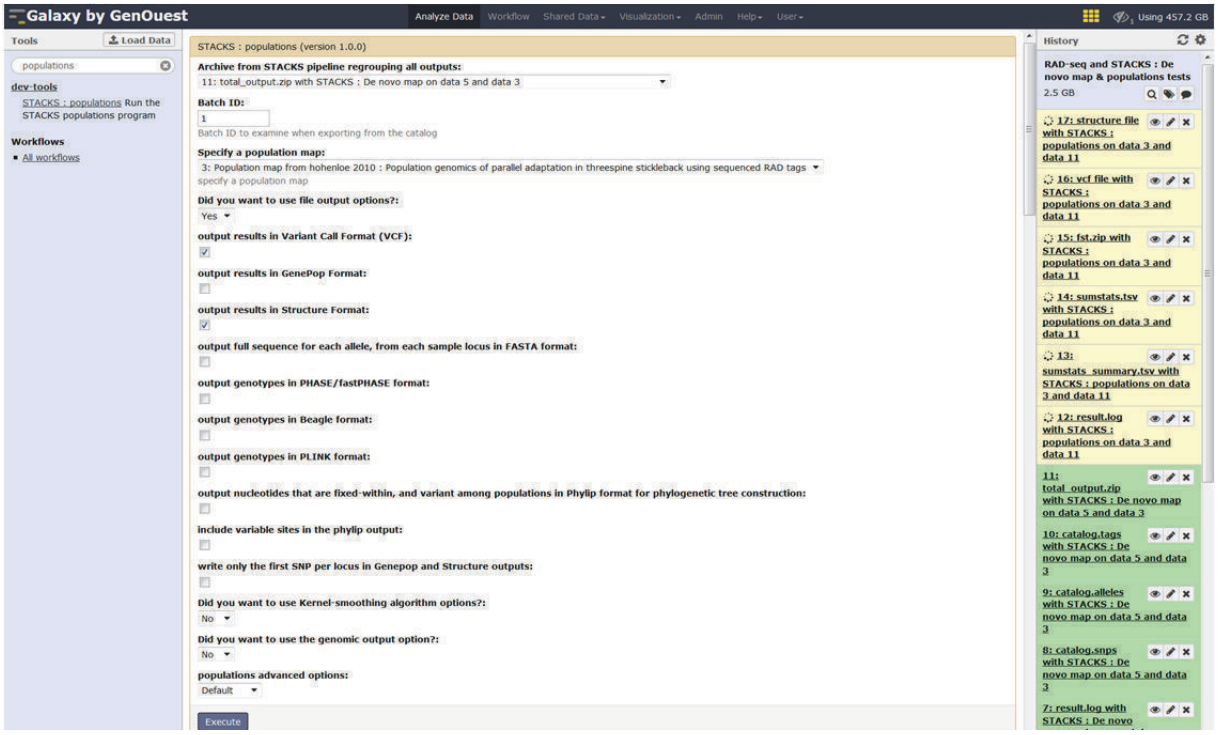
On choisira enfin le type de sortie : "compressed all outputs"



Une fois le job terminé, nous pouvons consulter les fichiers "*result.log*" et "*catalog.**" (* couvrant trois types de fichiers déjà mentionnés dans la première partie consacrée à la détection de SNP, à savoir *snps*, *alleles* et *tags*).



Afin de spécifier plus d'options et de pouvoir filtrer les résultats, il est possible de ré exécuter le dernier module de ***Stacks : de novo map***, à savoir ***populations*** sur l'archive "*total_output.zip*" générée dans l'étape précédente.



Il existe différentes options avancées. Il est notamment possible de préciser :

File output options :

-VCF

-Genepop

-Structure

-FASTA

-PHASE/fastPHASE

-Beagle

-PLINK

-Phylip

Kernel-smoothing algorithm options :

-le lissage des informations obtenues en moyennant les valeurs dans une fenêtre de taille en paires de base définie. si un génome de reference est utilisé

Genomic output options :

-l'export de chaque position nucléotidique (polymorphe ou non) de tous les individus d'une population dans un fichier. Il faudra alors préciser l'enzyme de restriction utilisée

Population advanced options :

-l'utilisation d'une whitelist. Dans ce cas, un fichier text constitué d'une colonne reprenant les Stack_ID à considérer pour l'analyse sera donné en entrée.

-l'utilisation d'une blacklist. Dans ce cas, un fichier text constitué d'une colonne reprenant les Stack_ID à ne pas considérer pour l'analyse sera donné en entrée.

-le pourcentage minimum d'individus par population pour pouvoir considérer le locus dans l'analyse

-le nombre minimum de populations dans lesquels le locus doit être présent pour le considérer dans l'analyse

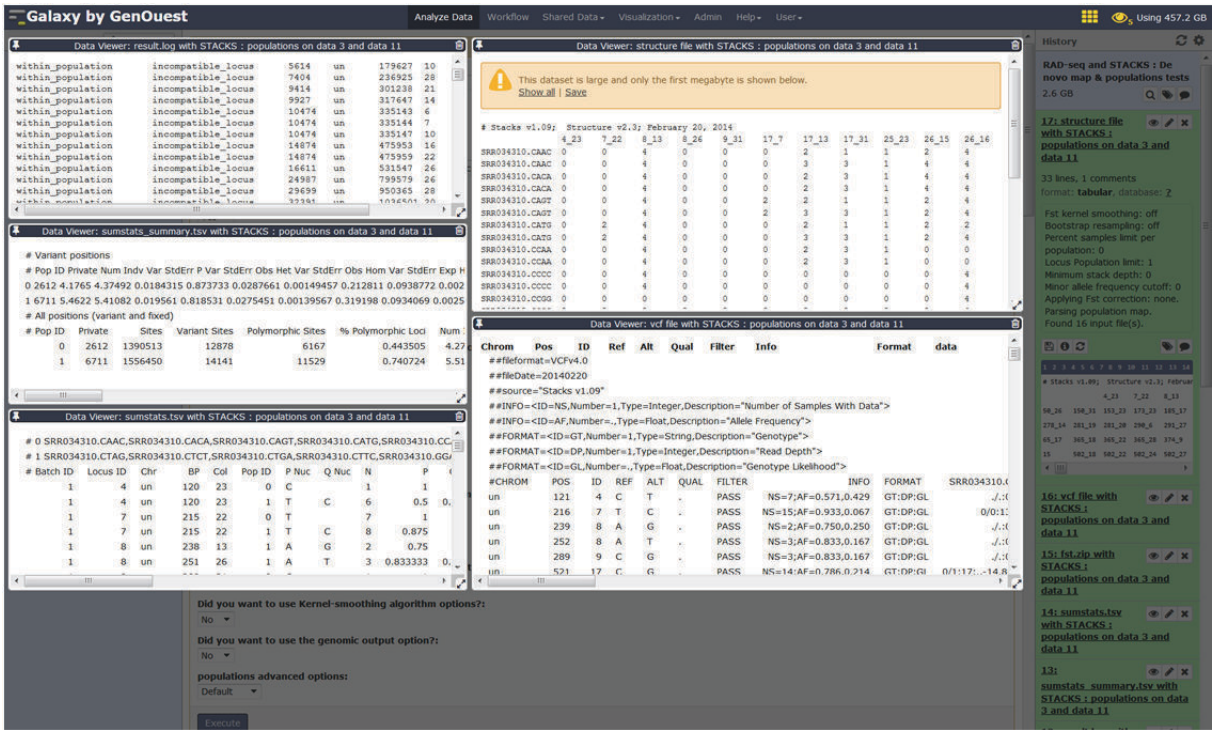
-la profondeur minimum d'un "stack" par individu à un locus donné

-une fréquence d'allèle alternatif / minoritaire minimum pour calculer un Fst au locus considéré

-un type de correction à appliquer aux valeurs de Fst

-un seuil de p-valeur pour conserver les valeurs de Fst

-d'effectuer un ré-échantillonnage par bootstrap en précisant la précision et le nombre. Attention, cela n'est applicable qu'à des données initialement mappées contre un génome de référence (via l'utilisation de l'outil ***Reference_map***).



Plusieurs types de fichiers sont générés :

1. Le fichier result.log

Il reprend les informations concernant les loci incompatible au sein de chacune des populations puis entre les populations.

NB : Pour consulter la sortie standard dans laquelle l'outil a écrit lors du déroulement du job, il faut cliquer sur le "*i*" d'un des jeux de données généré par l'outil puis sur "*stdout*".

Tool: STACKS : populations	
Name:	result.log with STACKS : populations on data 14 and data 19
Created:	Fri Aug 8 13:56:58 2014 (UTC)
Filesize:	6.3 KB
Dbkey:	?
Format:	txt
Galaxy Tool Version:	1.0.0
Tool Version:	
Tool Standard Output:	stdout
Tool Standard Error:	stderr
Tool Exit Code:	0
API ID:	2f4933c7d721d5e8
Full Path:	/omaha-beach/galaxy/58/database/files/000/dataset_224.dat
Job Command-Line:	python /opt/shed_tools/dev-galaxy.genouest.org/repos/cmonjeau/stacks_toolsuit beach/galaxy/58/database/files/000/dataset_180.dat --vcf true --genepop false -- beach/galaxy/58/database/files/000/dataset_225.dat --s /omaha-beach/galaxy/5 beach/galaxy/58/database/files/000/dataset_228.dat --os=/omaha-beach/galaxy beach/galaxy/58/database/tmp

2. Le fichier batch_X.sumstats.tsv

Batch ID	The batch identifier for this data set.
Locus ID	Catalog locus identifier.
Chromosome	If aligned to a reference genome.

Basepair If aligned to a reference genome. This is the alignment of the whole catalog locus. The exact basepair reported is aligned to the location of the RAD site (depending on whether alignment is to the positive or negative strand).
Column The nucleotide site within the catalog locus.
Population ID The ID supplied to the populations program, as written in the population map file.
P Nucleotide The most frequent allele at this position in this population.
Q Nucleotide The alternative allele.
Number of Individuals Number of individuals sampled in this population at this site.
P Frequency of most frequent allele.
Observed Heterozygosity The proportion of individuals that are heterozygotes in this population.
Observed Homozygosity The proportion of individuals that are homozygotes in this population.
Expected Heterozygosity Heterozygosity expected under Hardy-Weinberg equilibrium.
Expected Homozygosity Homozygosity expected under Hardy-Weinberg equilibrium.
pi An estimate of nucleotide diversity.
Smoothed pi A weighted average of p depending on the surrounding 3s of sequence in both directions.
Smoothed pi P-value If bootstrap resampling is enabled, a p-value ranking the significance of p within this population.
FIS The inbreeding coefficient of an individual (I) relative to the subpopulation (S).
Smoothed FIS A weighted average of FIS depending on the surrounding 3s of sequence in both directions.
Smoothed FIS P-value If bootstrap resampling is enabled, a p-value ranking the significance of FIS within this population.
Private allele True (1) or false (0), depending on if this allele is only occurs in this population.

1 SRR034310.CAAC,SRR034310.CACA,SRR034310.CAGT,SRR034310.CATG,SRR034310.CCAA,SRR034310.CCCC,SRR034310.CCGG,SRR034310.CCTT
2 SRR034310.CTAG,SRR034310.CTCT,SRR034310.CTGA,SRR034310.CTTC,SRR034310.GGAA,SRR034310.GGCC,SRR034310.GGGG,SRR034310.GGTT

# Batch ID	Locus ID	Chr	BP	Col	Pop ID	P Nuc	Q Nuc	N	P	Obs Het	Obs Hom	Exp Het	Exp Hom	Pi	Smoothed Pi	Smoothed Pi P-value	Fis	Smoothed Fis	Smoothed Fis P-value	Private
1	1	un	20	19	1	G	T	7	0.928571	0.142857	0.857143	0.132653	0.867347	0.142857	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	1	un	20	19	2	G		8	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	4	un	104	7	1	T	G	1	0.5	1	0	0.5	0.5	1	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	4	un	104	7	2	T		1	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	4	un	108	11	1	G	A	1	0.5	1	0	0.5	0.5	1	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	4	un	108	11	2	A		1	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	4	un	111	14	1	G	A	1	0.5	1	0	0.5	0.5	1	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	4	un	111	14	2	A		1	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	4	un	115	18	1	T	C	1	0.5	1	0	0.5	0.5	1	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	4	un	115	18	2	C		1	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	4	un	117	20	1	C	A	1	0.5	1	0	0.5	0.5	1	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	4	un	117	20	2	A		1	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	15	un	474	25	1	A		6	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	15	un	474	25	2	A	G	7	0.928571	0.142857	0.857143	0.132653	0.867347	0.142857	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	17	un	525	12	1	G		6	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	17	un	525	12	2	G	A	5	0.8	0.4	0.6	0.32	0.68	0.355556	0.0000000000	0	-0.1250000000	0.0000000000	0	1
1	19	un	608	31	1	T	G	3	0.5	1	0	0.5	0.5	0.6	0.0000000000	0	-0.6666666667	0.0000000000	0	0
1	19	un	608	31	2	G	T	3	0.666667	0.666667	0.333333	0.444444	0.555556	0.533333	0.0000000000	0	-0.2500000000	0.0000000000	0	0
1	20	un	625	16	1	C	A	1	0.5	1	0	0.5	0.5	1	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	20	un	625	16	2	A		5	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	20	un	634	25	1	G	T	2	0.75	0.5	0.5	0.375	0.625	0.5	0.0000000000	0	0.0000000000	0.0000000000	0	1
1	20	un	634	25	2	G		4	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	20	un	636	27	1	G	A	2	0.75	0.5	0.5	0.375	0.625	0.5	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	20	un	636	27	2	G	A	5	0.6	0.4	0.6	0.48	0.52	0.533333	0.0000000000	0	0.2500000000	0.0000000000	0	0
1	24	un	755	18	1	T	C	7	0.571429	0.571429	0.428571	0.489796	0.510204	0.527473	0.0000000000	0	-0.0833333333	0.0000000000	0	0
1	24	un	755	18	2	T	C	8	0.625	0.5	0.5	0.46875	0.53125	0.5	0.0000000000	0	0.0000000000	0.0000000000	0	0
1	28	un	881	16	1	G		5	1	0	1	0	1	0	0.0000000000	0	0.0000000000	0.0000000000	0	0

Accompagné du fichier sumstats_summary.tsv (disposé en 2 parties pour faciliter la visualisation) :

Variant positions

Pop ID Private Num Indv Var StdErr P Var StdErr Obs Het Var StdErr Obs Hom Var StdErr Exp Het Var StdErr Exp Hom Var StdErr Pi Var StdErr Fis Var StdErr

0 2665 4.15392 4.32523 0.0184255 0.874743 0.0285654 0.00149739 0.217513 0.0959329 0.00274409 0.782487 0.0959329 0.00274409 0.162009 0.0369834

1 6654 5.40317 5.47518 0.0197589 0.82038 0.0275747 0.00140223 0.31146 0.0922011 0.00256408 0.68854 0.0922011 0.00256408 0.239568 0.0303993 0.0

All positions (variant and fixed)

# Pop ID	Private	Sites	Variant Sites	Polymorphic Sites	% Polymorphic Loci	Num Indv	Var	StdErr	P	Var
0	2665	1395856	12740	6086	0.436005	4.24817	4.80505	0.00185536	0.998857	0.000402586
1	6654	1551311	14024	11359	0.732219	5.52033	6.39777	0.00203079	0.998376	0.000538288

0.0017038 0.837991 0.0369834 0.0017038 0.21319 0.0792028 0.00249336 -0.00576392 0.0445887 0.0184255

0014723 0.760432 0.0303993 0.0014723 0.302335 0.0715341 0.0022585 -0.0157131 0.0580463 0.0197589

StdErr	Obs Het	Var	StdErr	Obs Hom	Var	StdErr	Exp Het	Var	StdErr	Exp Hom	Var	StdErr	Pi	Var	StdErr	Fis	Var	StdErr
1.69828e-05	0.00198525	0.00130339	3.05574e-05	0.998015	0.00130339	3.05574e-05	0.00147866	0.000574891	2.02942e-05	0.998521	0.000574891	2.02942e-05	0.00194579	0.00113386	2.8501e-05	-5.26073e-05	0.000407231	0.00185536
1.86276e-05	0.00281563	0.00170248	3.31277e-05	0.997184	0.00170248	3.31277e-05	0.00216572	0.000788939	2.25513e-05	0.997834	0.000788939	2.25513e-05	0.00273314	0.00146548	3.07356e-05	-0.000142048	0.000526918	0.00203079

3. Le fichier batch_X.fst_Y-Z.tsv

Batch ID — The batch identifier for this data set.

Locus ID — Catalog locus identifier.

Population ID 1 — The ID supplied to the populations program, as written in the population map file.

Population ID 2 — The ID supplied to the populations program, as written in the population map file.

Chromosome — If aligned to a reference genome.

Basepair — If aligned to a reference genome. This is the alignment of the whole catalog locus. The exact basepair reported is aligned to the location of the RAD site (depending on whether alignment is to the positive or negative strand).

Column — The nucleotide site within the catalog locus.

Overall pi — An estimate of nucleotide diversity across the two populations.

FST — A measure of population differentiation.

FET p-value — P-value describing if the FST measure is statistically significant according to Fisher's Exact Test.

Odds Ratio — Fisher's Exact Test odds ratio

CI High — Fisher's Exact Test confidence interval.

CI Low — Fisher's Exact Test confidence interval.

LOD Score — Logarithm of odds score.

Expected Heterozygosity — Heterozygosity expected under Hardy-Weinberg equilibrium.

Expected Homozygosity — Homozygosity expected under Hardy-Weinberg equilibrium.

Corrected FST — FST with either the FET p-value, or a window-size or genome size Bonferroni correction.

Smoothed FST — A weighted average of FST depending on the surrounding 3s of sequence in both directions.

Smoothed FST P-value — If bootstrap resampling is enabled, a p-value ranking the significance of FST within this pair of populations.

# Batch ID	Locus ID	Pop 1 ID	Pop 2 ID	Chr	BP	Column	Overall Pi	Fst	Fisher's P	Odds Ratio	CI Low	CI High	LOD	Corrected Fst	Smoothed Fst	AMOVA Fst	Corrected AMOVA Fst	Smoothed AMOVA Fst
1	1	0	1	un	20	19	0.0666667	0.0758293839	0.466667	0.411765	0.0337135	5.02915	0.385351	0.0758293839	0.0000000000	0.0394088670	0.0394088670	0.0000000000
1	4	0	1	un	104	7	0.5	0.0000000000	0.5	0.333333	0.0166964	6.65479	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000
1	4	0	1	un	108	11	0.5	0.0000000000	0.5	3	0.150268	59.8931	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000
1	4	0	1	un	111	14	0.5	0.0000000000	0.5	3	0.150268	59.8931	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000
1	4	0	1	un	115	18	0.5	0.0000000000	0.5	3	0.150268	59.8931	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000
1	4	0	1	un	117	20	0.5	0.0000000000	0.5	3	0.150268	59.8931	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000
1	15	0	1	un	474	25	0.0769231	-0.0764331210	1	1.85714	0.149963	22.9989	0.268845	-0.0764331210	0.0000000000	0.0342857143	0.0342857143	0.0000000000
1	17	0	1	un	525	12	0.17316	0.1675675676	0.194805	4.33333	0.386277	48.6122	0.636822	0.1675675676	0.0000000000	0.1200000000	0.1200000000	0.0000000000
1	19	0	1	un	608	31	0.530303	-0.0685714286	0.621212	2	0.194034	20.6149	0.30103	-0.0685714286	0.0000000000	0.0285714286	0.0285714286	0.0000000000
1	20	0	1	un	625	16	0.166667	0.8695652174	0.166667	11	0.64645	187.176	1.04139	0.8695652174	0.0000000000	0.4545454545	0.4545454545	0.0000000000
1	20	0	1	un	634	25	0.166667	0.4705882353	0.333333	0.222222	0.0153298	3.22136	0.653213	0.4705882353	0.0000000000	0.1818181818	0.1818181818	0.0000000000
1	20	0	1	un	636	27	0.494505	-0.0705882353	1	2	0.149615	26.7353	0.30103	-0.0705882353	0.0000000000	0.0200000000	0.0200000000	0.0000000000
1	24	0	1	un	755	18	0.496552	-0.0308056872	1	0.8	0.184946	3.46046	0.09691	-0.0308056872	0.0000000000	0.0029761905	0.0029761905	0.0000000000
1	28	0	1	un	881	16	0.0769231	-0.1818181818	1	1.375	0.110603	17.0939	0.138303	-0.1818181818	0.0000000000	0.0250000000	0.0250000000	0.0000000000
1	31	0	1	un	977	16	0.198413	-0.0567741935	0.238095	3.71429	0.365843	37.71	0.569875	-0.0567741935	0.0000000000	0.0900000000	0.0900000000	0.0000000000
1	32	0	1	un	1009	16	0.0666667	-0.0663507109	1	1.875	0.153645	22.8815	0.273001	-0.0663507109	0.0000000000	0.0301724138	0.0301724138	0.0000000000
1	43	0	1	un	1371	26	0.268421	-0.0700876095	0.242105	3.6	0.336807	38.479	0.556303	-0.0700876095	0.0000000000	0.1176470588	0.1176470588	0.0000000000
1	44	0	1	un	1391	14	0.253968	0.1108870968	0.284982	0.2	0.0179743	2.2254	0.69897	0.1108870968	0.0000000000	0.0703125000	0.0703125000	0.0000000000
1	52	0	1	un	1643	10	0.212308	-0.1133069829	0.261538	3.14286	0.306001	32.2795	0.497325	-0.1133069829	0.0000000000	0.0815217391	0.0815217391	0.0000000000

Pop 2 ID	Chr	BP	Column	Overall Pi	Fst	Fisher's P	Odds Ratio	CI Low	CI High	LOD	Corrected Fst	Smoothed Fst	AMOVA Fst	Corrected AMOVA Fst	Smoothed AMOVA Fst	Smoothed AMOVA Fst P-value	Window SNP Count
1	un	20	19	0.0666667	0.0758293839	0.466667	0.411765	0.0337135	5.02915	0.385351	0.0758293839	0.0000000000	0.0394088670	0.0394088670	0.0000000000	0	0
1	un	104	7	0.5	0.0000000000	0.5	0.333333	0.0166964	6.65479	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000	0	0
1	un	108	11	0.5	0.0000000000	0.5	3	0.150268	59.8931	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000	0	0
1	un	111	14	0.5	0.0000000000	0.5	3	0.150268	59.8931	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000	0	0
1	un	115	18	0.5	0.0000000000	0.5	3	0.150268	59.8931	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000	0	0
1	un	117	20	0.5	0.0000000000	0.5	3	0.150268	59.8931	0.477121	0.0000000000	0.0000000000	0.3333333333	0.3333333333	0.0000000000	0	0
1	un	474	25	0.0769231	-0.0764331210	1	1.85714	0.149963	22.9989	0.268845	-0.0764331210	0.0000000000	0.0342857143	0.0342857143	0.0000000000	0	0
1	un	525	12	0.17316	0.1675675676	0.194805	4.33333	0.386277	48.6122	0.636822	0.1675675676	0.0000000000	0.1200000000	0.1200000000	0.0000000000	0	0
1	un	608	31	0.530303	-0.0685714286	0.621212	2	0.194034	20.6149	0.30103	-0.0685714286	0.0000000000	0.0285714286	0.0285714286	0.0000000000	0	0
1	un	625	16	0.166667	0.8695652174	0.166667	11	0.64645	187.176	1.04139	0.8695652174	0.0000000000	0.4545454545	0.4545454545	0.0000000000	0	0
1	un	634	25	0.166667	0.4705882353	0.333333	0.222222	0.0153298	3.22136	0.653213	0.4705882353	0.0000000000	0.1818181818	0.1818181818	0.0000000000	0	0
1	un	636	27	0.494505	-0.0705882353	1	2	0.149615	26.7353	0.30103	-0.0705882353	0.0000000000	0.0200000000	0.0200000000	0.0000000000	0	0
1	un	755	18	0.496552	-0.0308056872	1	0.8	0.184946	3.46046	0.09691	-0.0308056872	0.0000000000	0.0029761905	0.0029761905	0.0000000000	0	0
1	un	881	16	0.0769231	-0.1818181818	1	1.375	0.110603	17.0939	0.138303	-0.1818181818	0.0000000000	0.0250000000	0.0250000000	0.0000000000	0	0
1	un	977	16	0.198413	-0.0567741935	0.238095	3.71429	0.365843	37.71	0.569875	-0.0567741935	0.0000000000	0.0900000000	0.0900000000	0.0000000000	0	0
1	un	1009	16	0.0666667	-0.0663507109	1	1.875	0.153645	22.8815	0.273001	-0.0663507109	0.0000000000	0.0301724138	0.0301724138	0.0000000000	0	0
1	un	1371	26	0.268421	-0.0700876095	0.242105	3.6	0.336807	38.479	0.556303	-0.0700876095	0.0000000000	0.1176470588	0.1176470588	0.0000000000	0	0
1	un	1391	14	0.253968	0.1108870968	0.284982	0.2	0.0179743	2.2254	0.69897	0.1108870968	0.0000000000	0.0703125000	0.0703125000	0.0000000000	0	0
1	un	1643	10	0.212308	-0.1133069829	0.261538	3.14286	0.306001	32.2795	0.497325	-0.1133069829	0.0000000000	0.0815217391	0.0815217391	0.0000000000	0	0

4. Les fichiers de sortie optionnels

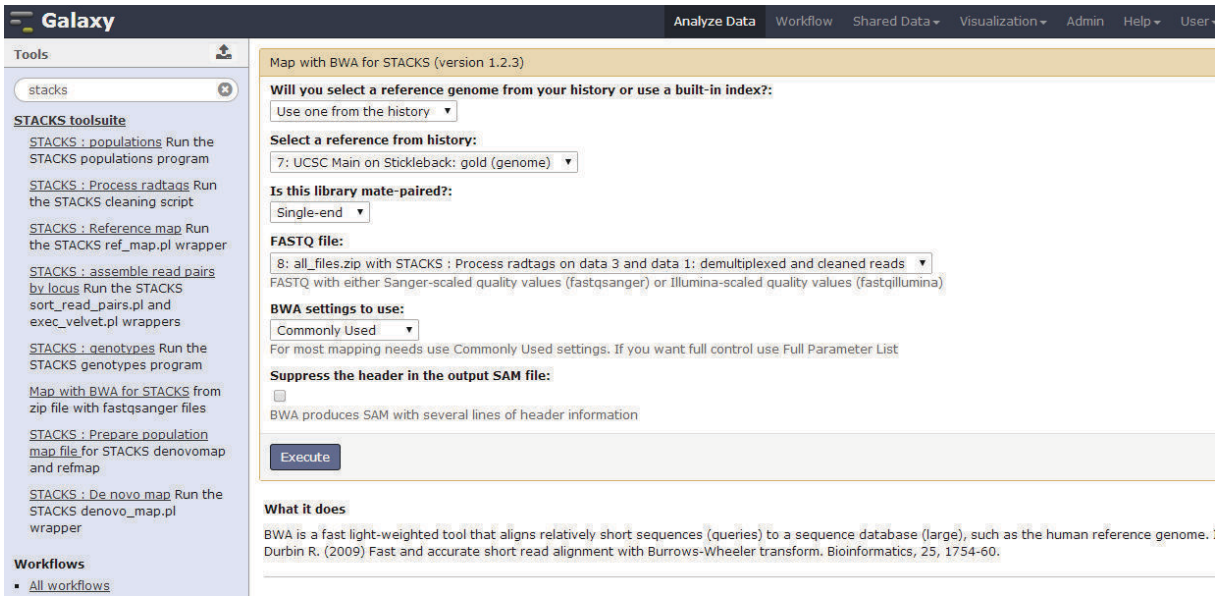
Dans la capture d'écran précédente sont par exemple présentés des jeux de données au format :

- VCF, pour être vue par exemple sur des visualisateurs de génome comme IGV ou Trackster (outil embarqué dans Galaxy)

- structure, pour servir de fichier d'entrée du logiciel Structure

5. Utilisation d'un génome de référence

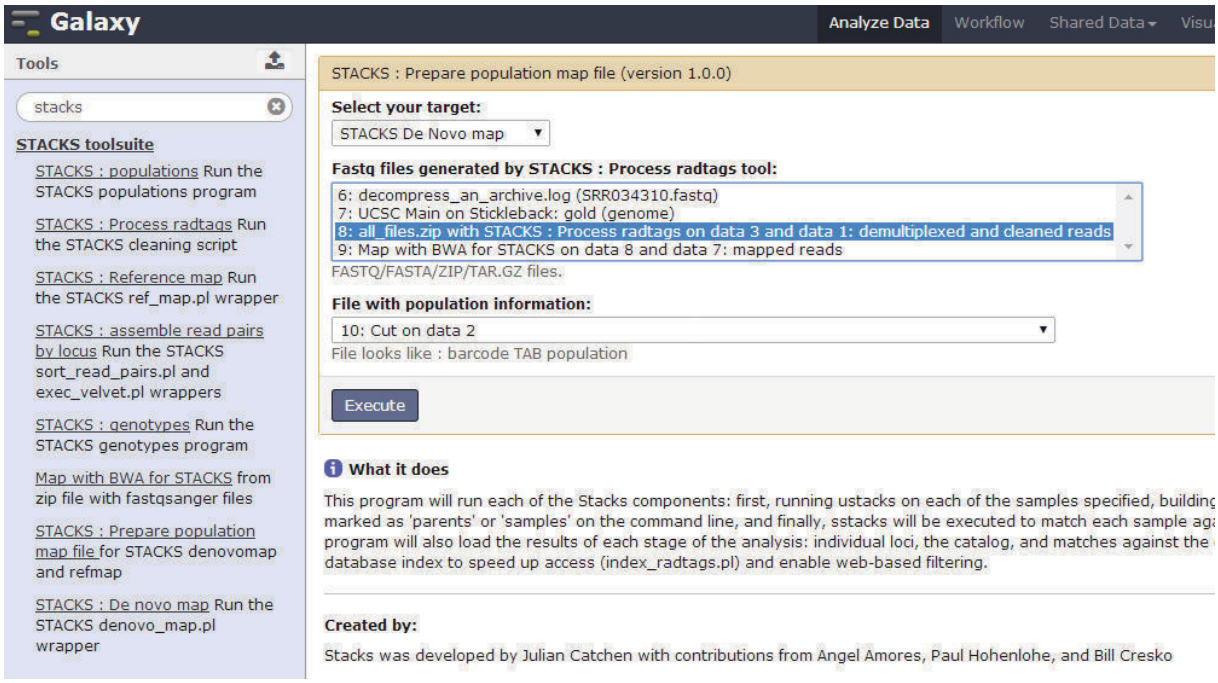
Après avoir récupéré l'archive générée précédemment par ***STACKS : Process Radtags***, le fichier à 2 colonnes contenant les informations de population, et avoir téléchargé un génome de référence via l'UCSC (Stickelback gold, sequence), la première étape consiste à aligner les lectures démultiplexées sur le génome. Pour cela, nous utiliserons BWA à travers un outil spécialement réalisé pour STACKS : Map with BWA for STACKS.

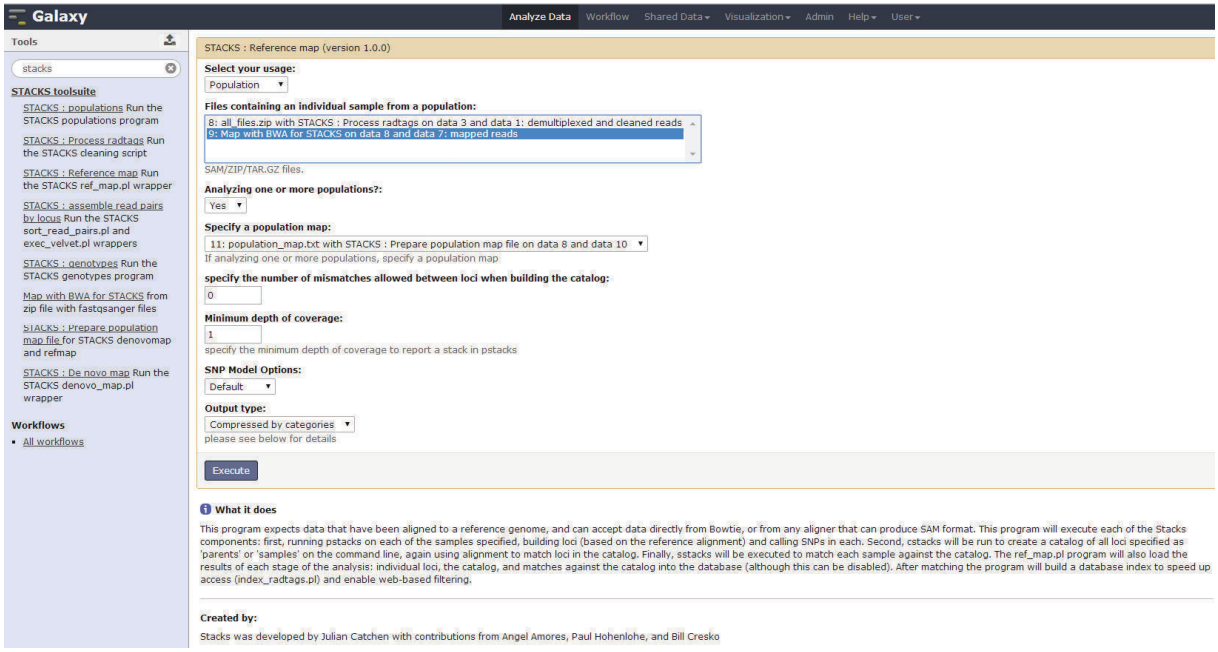


La spécificité de cet outil est de pouvoir générer une archive contenant les fichiers d'alignement au format *SAM* portant le même nom que dans le fichier "*population map*".

Afin de générer un fichier de type population map possédant et mappant les bonnes informations, nous proposons l'utilisation de l'outil STACKS : Prepare population map file for STACKS denovomap and refmap. Il prend en entrée les fichiers fastq demultiplexés (organisés ou non dans une archive) ainsi qu'un fichier permettant de lier les barcodes aux populations, ici Details_Barcode_Population_SRR034310.txt présentant exactement 2 colonnes, la première colonne contenant les barcodes, la seconde le nom de la population.

Nous pouvons ensuite exécuter *STACKS : Prepare population map file* en utilisant le fichier à 2 colonne généré à l'étape précédente pour donner la relation individu / population.





Références :

http://evomics.org/learning/genomics/stacks/ : Cours de Julian sur Evomics. Voir en particuliers les références mentionnées. Je les reprends ici pour information :

Core readings for the lecture and workshop

Amores, A., et al. 2011. Genome evolution and meiotic maps by massively parallel DNA sequencing. Genetics 188:799-808.

Broman, K. W. 2010. Genetic map construction with R/qtl. Univ. Wisc. Technical Report #214.

Catchen, J. M. et al. 2011. Stacks: building and genotyping loci de novo from short-read sequences. G3: Genes, Genomes and Genetics 1; 171-182.

Davey, J. W., et al. 2011. Genome-wide genetic marker discovery and genotyping using next-generation sequencing. Nature Reviews Genetics 12:499-510.

Etter, P. D., et al. 2011. SNP Discovery and Genotyping for Evolutionary Genetics using RAD sequencing. in Molecular Methods in Evolutionary Genetics, Rockman, M., and Orgonogozo, V., eds. (in press).

Ekblom, R., and J. Galindo. 2010. Applications of next generation sequencing in molecular ecology of non-model organisms. Heredity 107:1-15.

Hohenlohe, P. A. et al. 2010. Population genomics of parallel adaptation in threespine stickleback using sequenced RAD tags. PLoS Genetics 6. 1-23.

NGS population genomics background, concepts and statistical considerations

Broman, K. W., et al. 2003. R/qtl: QTL mapping in experimental crosses. Bioinformatics 19:889-890.

Broman, K. W., and S. Sen. 2009. A Guide to QTL Mapping with R/qtl. Springer.

Gompert, Z., and C. A. Buerkle. 2011a. A hierarchical Bayesian model for next-generation population genomics. Genetics 187:903-917.

Gompert, Z., and C. A. Buerkle. 2011b. Bayesian estimation of genomic clines. Molecular Ecology 20:2111-2127.

Lynch, M. 2009. Estimation of allele frequencies from high-coverage genome-sequencing projects. Genetics 182:295-301.

Nielsen, R., et al. 2005. Genomic scans for selective sweeps using SNP data. Genome Research 15:1566-1575.

Hohenlohe, P. A., et al. 2010. Using population genomics to detect selection in natural populations: Key concepts and methodological considerations. International Journal of Plant Sciences 171:1059-1071.

Stapley, J., et al. 2010. Adaptation genomics: the next generation. Trends in Ecology and Evolution 25:705-712.

Luikart, G.,et al. 2003. The power and promise of population genomics: from genotyping to genome typing. Nature Reviews Genetics 4:981-994.

Nielsen, R., et al. 2011. Genotype and SNP calling from next-generation sequencing data. Nature Reviews Genetics 12:443-451.

Genetic mapping using RRL and RAD sequencing

Altshuler, D., et al. 2000. An SNP map of the human genome generated by reduced representation shotgun sequencing. Nature 407:513-516.

Baxter, S. W., et al. 2011. Linkage mapping and comparative genomics using next-generation RAD sequencing of a non-model organism. PLoS ONE 6:e19315.

Chutimanitsakun, Y., et al. 2011. Construction and application for QTL analysis of a Restriction Site Associated DNA (RAD) linkage map in barley. BMC Genomics 12: 1-13.

Gore, M. A., et al. 2009. A first-generation haplotype map of maize. Science 326:1115-1117.

RAD-seq genotyping methodology

Baird, N. A., et al. 2008. Rapid SNP discovery and genetic mapping using sequenced RAD markers. PLoS ONE 3:e3376.

Emerson, K. J., et al. 2010. Resolving postglacial phylogeography using high-throughput sequencing. Proceedings of the National Academy of Sciences 107:16196-16200.

Etter, P. D., et al. 2011. Local De Novo Assembly of RAD Paired-End Contigs Using Short Sequencing Reads. PLoS ONE 6:e18561

Hohenlohe, P. A., et al. 2011. Next-generation RAD sequencing identifies thousands of SNPs for assessing hybridization between rainbow and westslope cutthroat trout. Molecular Ecology Resources 11 Suppl 1:117-122.

Miller, M. R., et al. 2007. Rapid and cost-effective polymorphism identification and genotyping using restriction site associated DNA (RAD) markers. Genome Research 17:240-248.

Willing, E. M., et al. 2011. Paired-end RAD-seq for de novo assembly and marker design without available reference. Bioinformatics 27:2187-2193.

Other reduced representation library (RRL) methodologies

Andolfatto, P., et al. 2011. Multiplexed shotgun genotyping for rapid and efficient genetic mapping. Genome Research 21:610-617.

Elshire, R. J., et al. 2011. A Robust, Simple Genotyping-by-Sequencing (GBS) Approach for High Diversity Species. PLoS ONE 6:e19379.

Rigola, D., et al. 2009. High-Throughput Detection of Induced Mutations and Natural Variation Using KeyPoint™ Technology. PLoS ONE 4:e4761.

van Orsouw, N. J., et al. 2007. Complexity reduction of polymorphic sequences (CRoPS): a novel approach for large-scale polymorphism discovery in complex genomes. PLoS ONE 2:e1172.

van Tassell, C. P., et al. 2008. SNP discovery and allele frequency estimation by deep sequencing of reduced representation libraries. Nature Methods 5:247-252.

Useful links

1. Quality scores

1. http://en.wikipedia.org/wiki/FASTQ_format
2. http://en.wikipedia.org/wiki/Phred_quality_score
3. http://www.phrap.com/phred/
4. http://www.illumina.com/truseq/quality_101/quality_scores.ilmn

2. Basic Unix, R and PERL commands

1. http://mally.stanford.edu/~sr/computing/basic-unix.html
2. http://korflab.ucdavis.edu/Unix_and_Perl/
3. http://www.r-project.org/
4. http://cran.r-project.org/doc/manuals/R-intro.html
5. http://manuals.bioinformatics.ucr.edu/home/programming-in-r

3. Stacks download and tutorials

1. http://creskolab.uoregon.edu/stacks/

4. Great site for information on next gen sequencing

1. http://seqanswers.com/