

## QDD2

For Windows and Linux  
Version 2 (28 June 2011)

A user-friendly program to select microsatellite markers and design primers from large sequencing projects

**Emese Megléc<sup>1</sup> and Jean-François Martin<sup>2</sup>**

<sup>1</sup>Aix-Marseille Université, CNRS, IRD, UMR 6116 – IMEP, Equipe EGE Case 36, 3 Place Victor Hugo, 13331; Marseille Cedex 3, France

<sup>2</sup>Montpellier SupAgro, INRA, CIRAD, IRD, Centre de Biologie et de Gestion des Populations, Campus International de Baillarguet, CS30016, 34988 Montferrier-sur-Lez, France

emese.meglecz@univ-provence.fr

<http://www.up.univ-mrs.fr/Local/egee/dir/meglecz/QDD.html>

### **To cite QDD:**

Megléc, E. Costedoat, C., Dubut, V., Gilles, A., Malausa, T., Pech, N. and Martin J-F. 2010. QDD: a user-friendly program to select microsatellite markers and design primers from large sequencing projects. **Bioinformatics**, 26(3) 403–404

## Quick start for users familiar to QDD1:

### Installation of the program (almost the same as QDD1)

- Install Perl, ClustalW2, Primer3 (version 1; not yet with version 2), BLAST+ (QDD2 uses BLAST+). It will not run with the earlier BLAST programs)
- Copy the QDD2 files into a folder
- Make a project folder with the data files

### Running the program: (same as QDD1)

- Open a terminal
- Change directory to the folder containing QDD2
- Type perl pipeX.pl (X = 1, 2, 3, 4)
- Use the menu to set parameters
- Outfiles are found in subfolders (pipeX\_\*) of the project folder

### Changes between QDD1 et QDD2

- The minimum number of repetitions to define a microsatellite cannot be specified. All microsatellites with at least 5 repetitions of 2-6 base pair motifs are considered as target microsatellites. All 2-6 bp motifs with 3-4 repetitions are considered as nanosatellites. Homopolymers are detected from 5 repetitions.
- The output files of the pipe2 (detection and elimination of redundant sequences) are more informative than in QDD1: One file is produced per sequence type: consensus (xxx\_consensus.fas), unique (xxx\_unique.fas), multihit (xxx\_multihit\_css.fas; Critically Simple Sequences, possibly minisatellites), nohit (xxx\_nohit\_css.fas; Critically Simple Sequences), grouped (xxx\_gr.fas; flanking regions similar to other loci but similarity is under the user defined threshold)) + one file with similar sequences aligned and their consensus (xxx\_cons\_subs.fas).
- Pure micro- and nanosatellites are pooled into a compound microsatellite if the distance between them is equal or less than the longest of the two motifs. Homopolymers are never pooled with micro- or nanosatellites.
- For reads that have more than one target microsatellites, all possible combinations of the target microsatellites are defined as target region. QDD2, thus allows the design for one single target microsatellite or for more than one microsatellite in the target region.
- For each target microsatellite (at least 5 uninterrupted repetitions) primers are designed with 7 different designs:

Design	Homopolymer allowed in the flanking and in primer*	Other target microsatellite allowed in flanking region	Nanosatellite allowed in primer	Nanosatellite allowed in flanking (not in primer)	Target microsatellite can be compound
A	-	-	-	-	-
B	-	-	-	+	+
C	-	-	+	+	+
D	-	+	-	+	+
E	-	+	+	+	+
F	+	-	+	+	+
G	+	+	+	+	+

\*A separate primer3 parameter allows setting the maximum length of a homopolymer in the primer.

- Apart from the information already present in the QDD1 primer table it also contains
  - o Column marking polymorph target microsatellites (in consensus sequences)
  - o Column with the design type (A-G)
  - o Column with the best (lowest penalty for the best design type available) primer pair for each target region
  - o Column with the best (lowest penalty for the best design type available) primer pair for each sequence
- A pipe4 using bioperl is available to BLAST sequences with primer against GenBank (nt). The primer table is completed with information on the best hit to Genbank (if any) and the lineage of the organism of the hit (the subject sequence). This step is not sufficient for genomic comparisons, but users can spot out a contamination. DO NOT run pipe4 abusively since it can overcharge NCBI BLAST services. A good internet connection is necessary for this step.
- Each pipeX.pl produces a log file with all input parameters and summary statistics on the results.

## 1. Overview

Large scale sequencing has become affordable, therefore it is likely to replace rapidly microsatellite isolation involving cloning. Apart from cost- and labour-efficiency, access to a large number of sequences has two great advantages:

- (i) Microsatellite selection can be more stringent. Using only microsatellites that are not compound or interrupted, thus likely to follow a simple mutation model, provide markers that are more easily interpretable.
- (ii) Microsatellite amplification by PCR can be seriously affected by microsatellite and mobile element associations. The detection of large sequence clusters can suggest the presence of mobile elements, and thus eliminating microsatellites that are found in these clusters can increase the proportion of working primers compared to the total number of primers tested.

The original version of QDD (QDD1) aimed to select markers that are the most likely to give a clearly interpretable pattern. However, in the meantime NGS sequencing arrived to its adolescence, and allowed scientist to work at a genomic level even for non-model organism. The use of 50+ microsatellites markers for a species became a reality, and now, researchers need the most and not only the best out of their NGS data. QDD2 intends to follow this trend and provide primers also with less stringent design, but still giving information on the condition of the primer design.

QDD treats all bioinformatics steps from raw sequences until obtaining PCR primers: sorting sequences by tag, removing adapters/vectors, detection of microsatellites, detection of redundancy/possible mobile element association, selection of sequences with target microsatellites and primer design.

A user-friendly windows interface i-QDD2 is under development. The current version can be run both under Linux and Windows in an easy to use command line option.

## 2. Glossary

**Perfect (pure) microsatellite:** Microsatellite composed of one single motif of 2-6 bp length with no interruption. The minimum number of repetition is arbitrary set to 5.

**Nanosatellite:** 3-4 tandem repetition of a 2-6 bp motif.

**Homopolymer:** At least 5 tandem repetition of a single base.

**Compound microsatellite:** Pure micro- and nanosatellites are pooled into a compound microsatellite if the distance between them is equal or less than the longest of the two motifs. Homopolymers are never pooled with micro- or nanosatellites.

**Target microsatellite:** Pure or compound microsatellite with at least 5 uninterrupted repetitions of a 2-6 bp motif.

**Target region:** The region of the read that should be between the primers. There can be one or more target microsatellites in a target region.

**Genomic multicopies:** Loci present more than once in the genome. They can be either the results of duplication events or transposition.

**Flanking region:** The whole sequence apart from the target microsatellites. This simple definition can be applied, since the lengths of the reads are compatible with PCR, thus it is not necessary to define a maximum for length of a flanking region.

**Soft masking in BLAST:** BLAST prevents seeding (starting the alignment by a perfect match of a predefined length) in masked regions, but allows alignment extension through them if soft masking is applied.

**Tag:** A short DNA stretch added at the 5'-end of the DNA fragment to be sequenced for identification. Different tags can be added to DNA from different sources (e.g. species) and the pooled DNA is loaded on a non-fractionated PicoTiter plate, thus gaining space and quantities of reads. Sequences coming from different sources are identified according to their tag.

### 3. Installation

QDD is written in Perl and is run as a standalone application on Windows or Linux systems.

For both versions the following freely available programs should be installed in order to be able to run QDD:

**ActivePerl** (<http://www.activestate.com/activeperl/>)

**Bioperl** ( <http://www.bioperl.org/> ; It is only necessary for pipe4;

help for windows installation [http://www.bioperl.org/wiki/Installing\\_Bioperl\\_on\\_Windows](http://www.bioperl.org/wiki/Installing_Bioperl_on_Windows) )

**BLAST+** (<ftp://ftp.ncbi.nih.gov/blast/executables/blast+/> ; Use BLAST+ not BLAST)

**ClustalW** (<ftp://ftp.ebi.ac.uk/pub/software/clustalw2/>) Use clustalw2 and not formerly widely used clustalw1.83.

**Primer3** (<http://primer3.sourceforge.net/>) Use Primer3-1.1.4 version.

**3.1.** Install ActivePerl, BLAST, ClustalW2 and Primer3

Important: If you are working on MS Windows install Clustal2 using the msi file and keep the files within the folder selected during the installation process

If you are working on Linux install the package `csv_xs` (`sudo apt-get install libtext-csv-xs-perl`)

**3.2.** Untar and unzip QDD.tar.gz for Linux, extract QDD.rar (by WinRar) for windows

Put all files into one folder

**3.3.** Make a project folder for the input files

### 4. Description

QDD is composed of four parts. Each of them can be run separately.

#### 4.1 Sequence cleaning and microsatellite detection: pipe1.pl

Most of the steps do not take longer than a few minutes. If there are a million of sequences in the tag sorting step, it can take about 30 minutes.

##### 4.1.1.Input files

All input files must be in the project folder that does not contain other fasta files. The name of the input folder is set by the user in the parameters (see 4.2).

From here onwards we give the names of the output files for a run where the original input fasta file was named 'sample.fas' and put into a project folder 'datain' that is a subfolder within QDD. (data/sample.fas)

4.1.1.1. tag.fas (must be named 'tag.fas'; fasta file with all tag sequences; optional)

e.g.  
>MID1  
ACGAGTGCCT  
>MID2  
ACGCTCGACA

4.1.1.2. adapter.fas (must be named 'adapter.fas'; fasta file with all adapters/vectors that might be present in the sequences; optional but STRONGLY recommended where adapters apply)

4.1.1.3. fasta files from the sequencing project

There might be more than one file. The program deals with them one after the other.

The name of the fasta files can have any alphanumerical characters and underscore but not space and must have '.fas' extension (e.g. sample.fas). Everything in the definition line after '>' and before the first space is read as the sequence identifier. The identifier can have any alphanumerical characters and underscore. Replace all other characters by underscore.

e.g.  
>FVU26NR06DGVOE

```

ACGAGTGC GTGCCTAGCTAGCAGAATCACACACACACACACACACACACACACTATGTA
CTCTCCTTTGTGAAATACATACGACATGTGTACGTAAACAACACT
>FVU26NR06DIOGK
ACGAGTGC GTAAGGCCTAGCTAGCAGAATCGTTTCCTAATGATGCGCTTCCAAAACACT
CTCTGTGCGACTCTTTAACCTT
...

```

#### 4.1.2. Steps of pipe1.pl

- 4.1.2.1. It identifies and removes tags and writes one fasta file per tag with the tag free sequences (plus 1 file with sequences that did not have detectable tag). Optional.
- 4.1.2.2. Removes adaptors/vectors Optional. If adapter is not found at the beginning of the sequence, the sequence is removed.
- 4.1.2.3. Selects sequences longer or equal than the user-defined limit, since short sequences are likely to contain errors ( Gilles et al. 2011)
- 4.1.2.4. Selects sequences that contain microsatellites.
- 4.1.2.5. Output files are found in a pipe1\_XXX subfolder of your project folder, where XXX is a numerical identifier of the pipe1 run (different for each run) See explanation on the outfiles at point 6

#### 4.1.3. Parameters of pipe1.pl with default values

```

c:\WINDOWS\system32\cmd.exe - perl pipe1.pl
D:\QDD2_beta>
D:\QDD2_beta>
D:\QDD2_beta>
D:\QDD2_beta>
D:\QDD2_beta>
D:\QDD2_beta>
D:\QDD2_beta>perl pipe1.pl
*****
QDD version2 XX XX 2011
Emese Meglecz, Aix-Marseille University, Marseille, France
emese.meglecz@univ-provence.fr
Plesae, read the Documentation_QDD.pdf
*****
1 : Operating system (win/linux): win
2 : Project folder: datain
3 : Sort sequences by tag (YES=1/NO=0): 0
4 : Remove adaptors from sequences (YES=1/NO=0): 1
5 : Minimum sequence length: 80
6 : Pathway to BLAST executables: c:/Program files/NCBI/blast-2.2.25+/bin/
7 : Delete intermediate files (YES=1/NO=0): 1

Press enter if all of the settings are correct, or the number of the parameter i
f you wish to change the settings!

```

4.1.3.1. Operating system (win/linux): win

4.1.3.2. Input folder: e.g. datain

If the input folder is not the subfolder of the folder that contains the QDD scripts, the whole path should be specified. (e.g. c:\data). Only alphanumerical characters and underscore is allowed in the name

4.1.3.3. Sort sequences by tag (YES=1/NO=0): default = 1

If 1, QDD scans for tags defined in tag.fas, otherwise skips the tag sorting step

4.1.3.4. Remove adapter (YES=1/NO=0): default = 1

If 1, QDD removes vector/adaptor sequences. Attention! If adapters/vector/tags are not removed when they should, many sequences are unnecessarily eliminated by pipe2.pl (see 4.2.). Therefore skip these steps only if you are sure that you have a clean dataset.

4.1.3.5. Minimum sequence length: default = 80

Keeps sequences longer than 80 bp (without adapter and tag)

4.1.3.6. Pathway to BLAST executables: pathway to BLAST+ executables. Attention, executables are usually found in the /bin/ subfolder of BLAST+, therefore it must be included on the path (e.g. c:/Program files/NCBI/blast-2.2.25+/bin/)

4.1.3.7. Delete intermediate files (YES=1/NO=0): default = 1

If 1 keeps only important intermediate files, if it is set to 0 all intermediate files are kept (option used for troubleshooting, otherwise delete intermediate files is preferred)

## 4.2. Sequence similarity detection: pipe2.pl

This stage eliminates redundancy in the widest sense: copies of the same locus, sequences that potentially have more than one copy in the genome. The time of the run can vary from a few minutes to a few hours, and it depends on the number of sequences and the degree of redundancy (including intra genomic repetitions) of the data.

### 4.2.1. Input files

This stage can treat input files with up to 50 000 sequences in a single fasta file. Bigger files might be run, but it can be very long.

The input files were prepared by pipe1.pl and found in the pipe1\_XXX subfolder of the original input folder 'datain' (datain/pipe1\_XXX/sampe\_pipe2.fas).

If there are more than one subfolder starting by 'pipe1\_' in the project folder, the last in the alphabetical order will be checked for input files. (If you do not remane the folders, this should be the most recent folder produced by pipe1.pl)

If you want to run pipe2.pl on a fasta file, that is not produced by pipe1.pl create datain/pipe1\_9999999999 folder and put your input files named zzz\_pipe2.fas, where zzz can be replaced by a suite of any alphanumerical characters.

### 4.2.2. Steps of pipe2.pl

4.2.2.1. Detects sequence similarity by an all-against-all BLAST

4.2.2.2. Eliminates sequences that have more than 1 blast hit between the two same sequences (multihit, possibly minisatellites)

4.2.2.3. Eliminates sequences that did not have blast hit not even to themselves (nohit, possibly cryptically simple sequences)

4.2.2.4. Calculates pair wise identity along the whole flanking regions if similarity was detected by BLAST

4.2.2.5. Establishes contigs if pair wise similarity along the flanking region is higher than user-defined limit

4.2.2.6. Makes majority rule consensus sequences (consensus coefficient is user defined)

4.2.2.6. Check polymorphism in consensus sequences

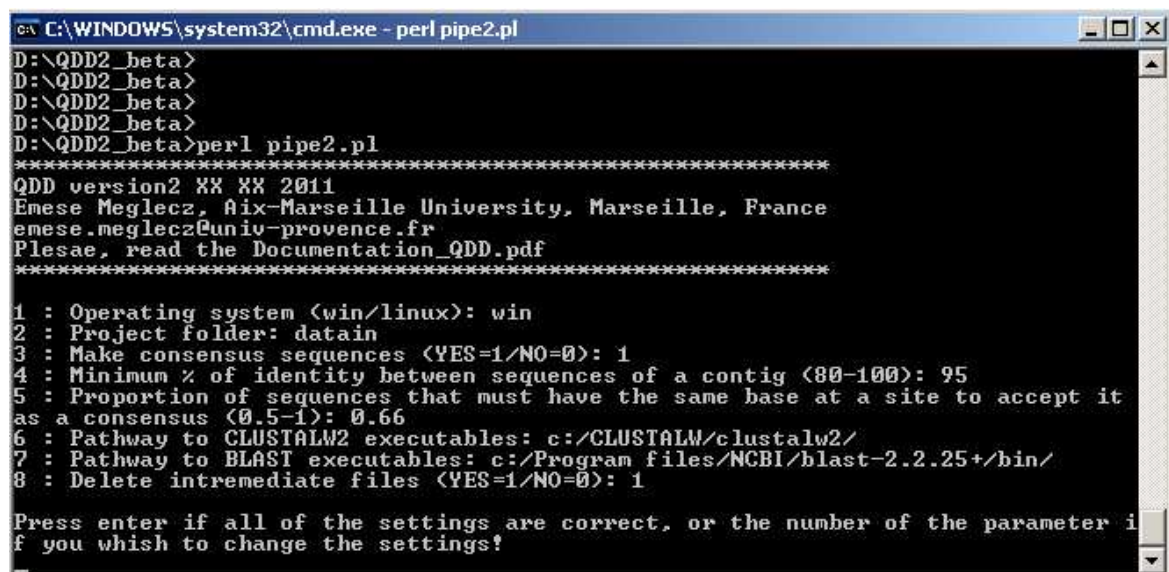
4.2.2.7. BLAST consensus sequences against sequences that have blast hits but not included in the contig (grouped sequences)

4.2.2.8. Selects consensus sequences that did not have hits in the previous BLAST

4.2.2.9. Prepares a file with selected consensus sequences and all original 'unique' sequences. This is placed into 'pipe2\_XXX' subfolder and will be the input file of pipe3.pl

4.2.2.10. Output files are found in a pipe2\_XXX subfolder of your project folder, where XXX is a numerical identifier of the pipe2 run (different for each run) See explanation on the outfiles at point 6

### 4.2.3. Parameters of pipe2.pl with default values



```
C:\WINDOWS\system32\cmd.exe - perl pipe2.pl
D:\QDD2_beta>
D:\QDD2_beta>
D:\QDD2_beta>
D:\QDD2_beta>
D:\QDD2_beta>perl pipe2.pl
*****
QDD version2 XX XX 2011
Emese Meglecz, Aix-Marseille University, Marseille, France
emese.meglecz@univ-provence.fr
Plesae, read the Documentation_QDD.pdf
*****
1 : Operating system (win/linux): win
2 : Project folder: datain
3 : Make consensus sequences (YES=1/NO=0): 1
4 : Minimum % of identity between sequences of a contig (80-100): 95
5 : Proportion of sequences that must have the same base at a site to accept it
as a consensus (0.5-1): 0.66
6 : Pathway to CLUSTALW2 executables: c:/CLUSTALW/clustalw2/
7 : Pathway to BLAST executables: c:/Program files/NCBI/blast-2.2.25+/bin/
8 : Delete intremediate files (YES=1/NO=0): 1

Press enter if all of the settings are correct, or the number of the parameter i
f you wish to change the settings!
```

4.2.3.1. Operating system (win/linux): win

#### 4.2.3.2. Input folder: datain

Must be the same as for pipe1.pl. This folder contains the subfolder pipe1\_XXX, with the input files of pipe2.pl

4.2.3.3. Make contigs and consensus sequences (if 1), else uses only sequences that had no Blast hit to other sequences. Unless you have a very large input file (more than 50 000 sequences) it is better to use option 1. If the run time is excessively long you can consider running option 0 (no contigs are prepared), and use only unique sequences in subsequent analyses.

4.2.3.4. Minimum % of identity between sequences of a contig (80%-100%): default = 95

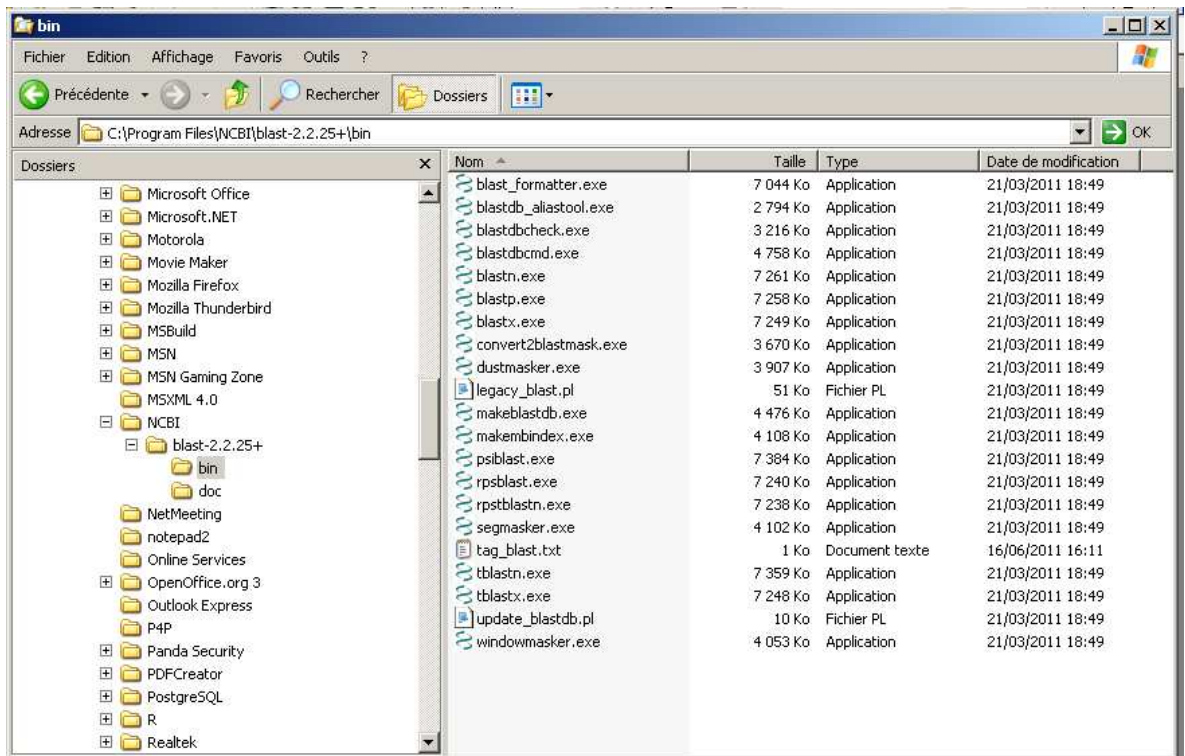
While making contigs a sequence is added to a contig if its flanking region similarity to at least one of the existing sequence in the contig is greater than 0.95

4.2.3.5. Proportion of sequences that must have the same base on the aligned site to accept it as a consensus: default = 0.66. Sequences of the contigs are aligned. For each site, a nucleotide is accepted as a consensus if it is present in more than 66% of the informative bases (not N) on that site. Otherwise N is put in the consensus sequence of the given site.

4.2.3.6. Pathway to BLAST+: e.g. C:/Program Files/NCBI/blast-2.2.25+/bin/ (including de bin folder that contains the executables)

4.2.3.7. Pathway to CLUSTALW: e.g. c:/CLUSTALW/clustalw2/

4.2.3.8. Delete intermediate files (YES=1/NO=0): default = 1



### 4.3. Microsatellite selection and primer design: pipe3.pl identifies target microsatellites: Pure or compound microsatellite with at least 5 uninterrupted repetitions of a 2-6 bp motif.

For each target microsatellite primers are designed with 7 different designs (A-G):

Design	Homopolymer allowed in the flanking and in primer*	Other target microsatellite allowed in flanking region	Nanosatellite allowed in primer	Nanosatellite allowed in flanking (not in primer)	Target microsatellite can be compound
A	-	-	-	-	-
B	-	-	-	+	+
C	-	-	+	+	+
D	-	+	-	+	+
E	-	+	+	+	+
F	+	-	+	+	+
G	+	+	+	+	+

\*A separate primer3 parameter allows to set the maximum length of a homopolymer in the primer.



### 4.3.1 Input files

The input files were prepared by pipe2.pl and found in the pipe2\_XXX subfolder of the original input folder 'datain' (datain/pipe2\_XXX/sampe\_pipe3.fas).

If there are more than one subfolder starting by 'pipe2\_' in the project folder, the last in the alphabetical order will be checked for input files. (If you do not remane the folders, this should be the most recent folder produced by pipe2.pl)

If you want to run pipe3.pl on a fasta file, that is not produced by pipe2.pl create datain/pipe2\_9999999999 folder and put your input files named zzz\_pipe3.fas, where zzz can be replaced by a suite of any alphanumeric characters.

### 4.3.2 Steps of pipe3.pl

4.3.2.1. Identifies all homopolymers, nano- and microsatellites

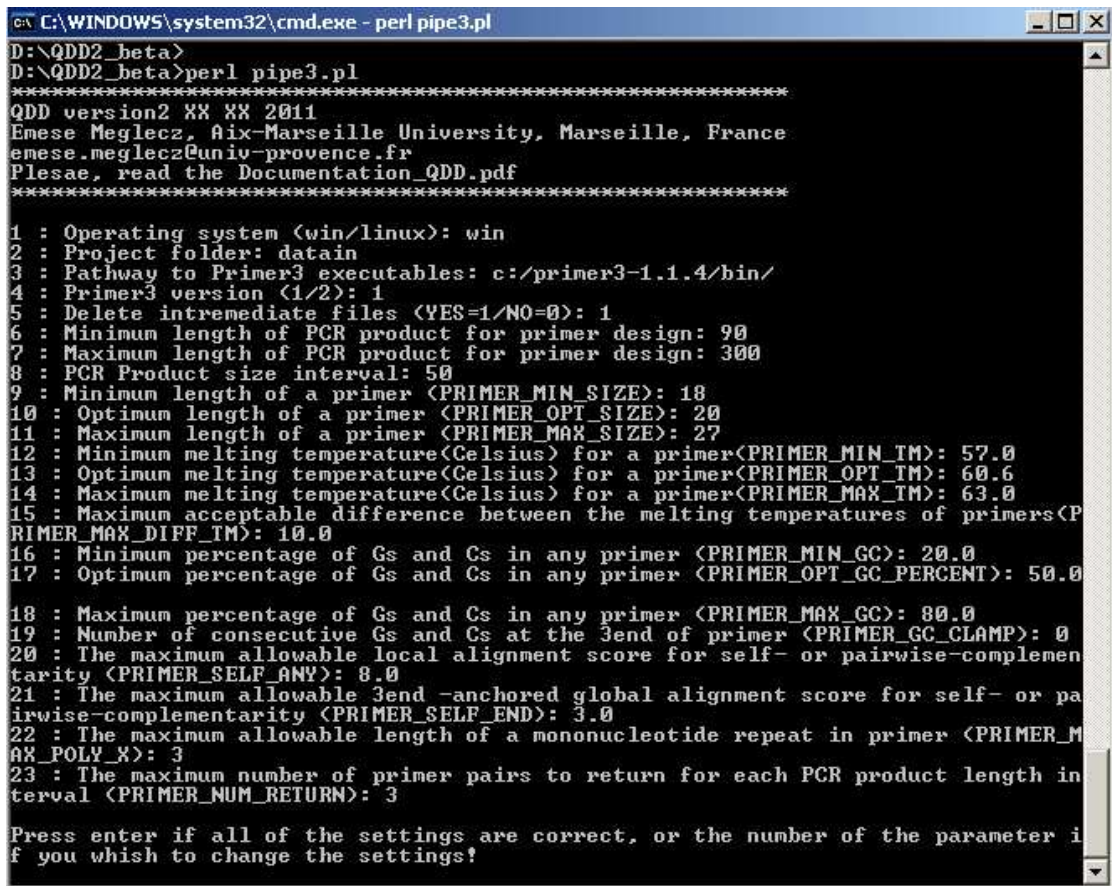
4.3.2.2. Pools nano- and microsatellites if the distance between them is lower or equal to the longest motif of the two neighbouring nano/microsatellites. They are treated as compound nano/microsatellites.

4.3.2.3. Identifies target microsatellites

4.3.2.4. Runs Primer3 for each user defined PCR product length interval and for each design. Most of the parameters for Primer3 can be set directly by a menu (see 4.3.3. for details) The target region (target microsatellite) and excluded region are defined automatically by QDD2 according to the more or less stringent design conditions (designs A-G).

All primer pairs and their descriptions are printed in a table, as well as the motif, length and position of the target microsatellite, condition of primer design and information of polymorphism of microsatellites of consensus sequences

### 4.3.3. Parameters of pipe3.pl with default values



```
C:\WINDOWS\system32\cmd.exe - perl pipe3.pl
D:\QDD2_beta>
D:\QDD2_beta>perl pipe3.pl
*****
QDD version2 XX XX 2011
Emese Meglecz, Aix-Marseille University, Marseille, France
emese.meglecz@univ-provence.fr
Plesae, read the Documentation_QDD.pdf
*****
1 : Operating system (win/linux): win
2 : Project folder: datain
3 : Pathway to Primer3 executables: c:/primer3-1.1.4/bin/
4 : Primer3 version (1/2): 1
5 : Delete intermediate files (YES=1/NO=0): 1
6 : Minimum length of PCR product for primer design: 90
7 : Maximum length of PCR product for primer design: 300
8 : PCR Product size interval: 50
9 : Minimum length of a primer (PRIMER_MIN_SIZE): 18
10 : Optimum length of a primer (PRIMER_OPT_SIZE): 20
11 : Maximum length of a primer (PRIMER_MAX_SIZE): 27
12 : Minimum melting temperature (Celsius) for a primer (PRIMER_MIN_TM): 57.0
13 : Optimum melting temperature (Celsius) for a primer (PRIMER_OPT_TM): 60.6
14 : Maximum melting temperature (Celsius) for a primer (PRIMER_MAX_TM): 63.0
15 : Maximum acceptable difference between the melting temperatures of primers (PRIMER_MAX_DIFF_TM): 10.0
16 : Minimum percentage of Gs and Cs in any primer (PRIMER_MIN_GC): 20.0
17 : Optimum percentage of Gs and Cs in any primer (PRIMER_OPT_GC_PERCENT): 50.0
18 : Maximum percentage of Gs and Cs in any primer (PRIMER_MAX_GC): 80.0
19 : Number of consecutive Gs and Cs at the 3end of primer (PRIMER_GC_CLAMP): 0
20 : The maximum allowable local alignment score for self- or pairwise-complementarity (PRIMER_SELF_ANY): 8.0
21 : The maximum allowable 3end-anchored global alignment score for self- or pairwise-complementarity (PRIMER_SELF_END): 3.0
22 : The maximum allowable length of a mononucleotide repeat in primer (PRIMER_MAX_POLY_X): 3
23 : The maximum number of primer pairs to return for each PCR product length interval (PRIMER_NUM_RETURN): 3

Press enter if all of the settings are correct, or the number of the parameter if you wish to change the settings!
```

4.3.3.1. Operating system (win/linux): win

4.3.3.2. Input folder: e.g. datain

Must be the same as for pipe1.pl. This folder contains the subfolder pipe2\_xxx, with the input files of pipe3.pl

4.3.3.3. Pathway to Primer3: e.g. c:/primer3-1.1.4/bin/



Path to Primer3 executables from the root; attention executables are found in the 'bin' subfolder of a folder that contains primer3

4.3.3.4. Primer3 version (At the moment QDD2 works only with primer3-1)

4.3.3.5. Deletes intermediate files (YES=1/NO=0): default = 1

4.3.3.6. Minimum length of PCR product for primer design (for Primer3): default = 90

4.3.3.7. Maximum length of PCR product for primer design (for Primer3): default = 320

4.3.3.8. Interval of length of PCR product for primer design (for Primer3): default = 50

Steps 4.3.3.6-8: Primer3 is run several times. Each time the desired PCR product size is set to a different interval to cover. As a default 90-140, 140-190, 190-240, 240-290, 290-320

**PRIMER3 internal parameters (for detailed explanation see Primer3 documentation):**

4.3.3.9. Minimum length of a primer (PRIMER\_MIN\_SIZE): default =18

4.3.3.10. Optimum length of a primer (PRIMER\_OPT\_SIZE): default =20

4.3.3.11. Maximum length of a primer (PRIMER\_MAX\_SIZE): default =27

4.3.3.12. Minimum melting temperature (Celsius) for a primer (PRIMER\_MIN\_TM): default =57.0

4.3.3.13. Optimum melting temperature (Celsius) for a primer (PRIMER\_OPT\_TM): default =60.0

4.3.3.14. Maximum melting temperature (Celsius) for a primer (PRIMER\_MAX\_TM): default =63.0

4.3.3.15. Maximum acceptable difference between the melting temperatures of primers (PRIMER\_MAX\_DIFF\_TM): default =10.0

4.3.3.16. Minimum percentage of Gs and Cs in any primer (PRIMER\_MIN\_GC): default =20.0

4.3.3.17. Optimum GC percent of primers (PRIMER\_OPT\_GC\_PERCENT): default =50.0

4.3.3.18. Maximum percentage of Gs and Cs in any primer (PRIMER\_MAX\_GC): default =80.0

4.3.3.19. Number of consecutive Gs and Cs at the 3' of primer (PRIMER\_GC\_CLAMP): default =0

4.3.3.20. The maximum allowable local alignment score for self- or pairwise-complementarity (PRIMER\_SELF\_ANY): default =8.00

4.3.3.21. The maximum allowable 3'-anchored global alignment score for self- or pairwise-complementarity (PRIMER\_SELF\_END): default =3.00

4.3.3.22. The maximum allowable length of a mononucleotide repeat in primer (PRIMER\_MAX\_POLY\_X): default =5

4.3.3.23. The maximum number of primer pairs to return for each PCR product length interval (PRIMER\_NUM\_RETURN): default =3

#### 4.4. BLAST sequences with primers to Genbnak: pipe4.pl

Pipe4.pl BLAST sequences with primer against GenBank (nt). The primer table produced by pipe3.pl is completed with information on the best hit to Genbank (if any) and the lineage of the organism from which the hit came from. This step is not sufficient for genomic comparisons, but users can spot out a contamination.!!!!!!!!!!!!!! DO NOT run pipe4 abusively since it can overcharge NCBI BLAST services !!!!!!!!!!!!!!!!!!!!!!! This step can take a lot of time and a good connection to internet is necessary to run it.

##### 4.4.1 Input files

The input files were prepared by pipe3.pl and found in the pipe3\_XXX subfolder of the original input folder 'datain' (datain/pipe3\_XXX/sample\_pipe3\_targets.fas; datain/pipe3\_XXX/sample\_pipe3\_primers.csv).

If there are more than one subfolder starting by 'pipe3\_' in the project folder, the last in the alphabetical order will be checked for input files. (If you do not rename the folders, this should be the most recent folder produced by pipe3.pl)

##### 4.1.2 Steps of pipe4.pl

4.4.2.1. BLAST sequences against the non-redundant nucleotide bank of NCBI (nt).

4.4.2.2. Identifies the taxonomic lineage of the organism of the best hit

4.4.2.3. Completes the primer table with information on the best hit and the taxonomy lineage of the organism.

##### 4.4.3. Parameters of pipe4.pl with default values

4.4.3.1. Operating system (win/linux): win

4.4.3.2. Input folder: e.g. datain

Must be the same as for pipe1.pl. This folder contains the subfolder pipe3, with the input files of pipe4.pl

4.4.3.3. Deletes intermediate files (YES=1/NO=0): default = 1

## 5. Running QDD2

### 5.1. Linux and Windows command line

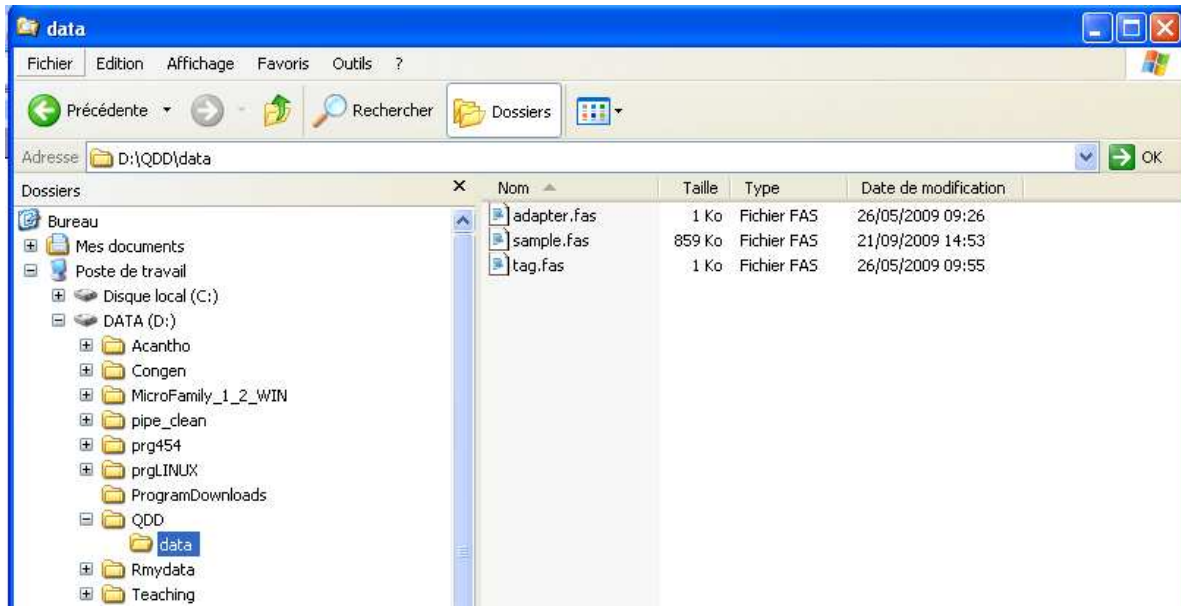
**5.1.1.** Put the input files of pipe1.pl into your project folder. The name of project folder can contain any alphanumerical character or underscore). All fasta files with the '.fas' extension are analysed, so make sure that the project folder contains only the files you want to analyse. **Do not use space in the name of the input files.**

Input files:

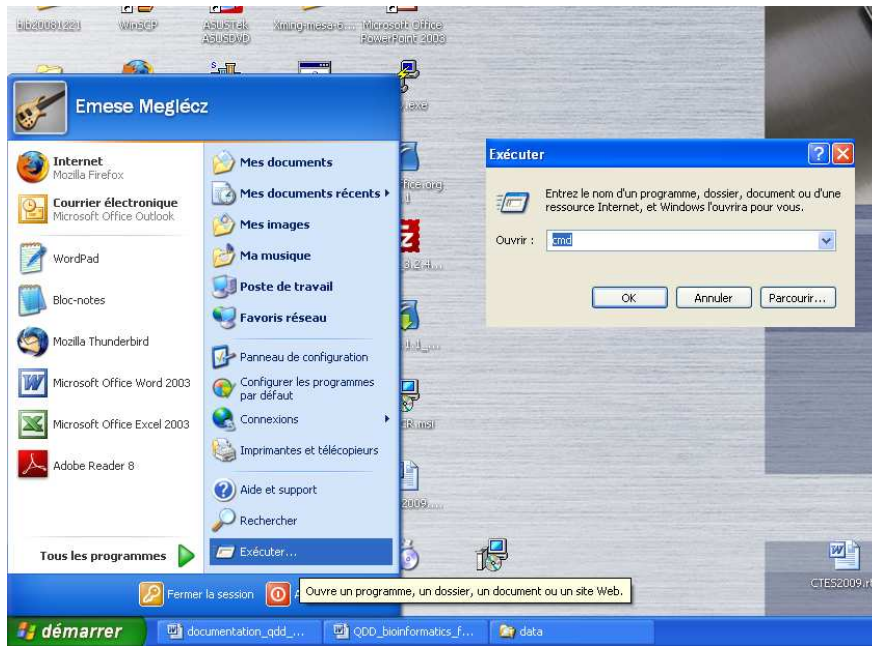
5.1.1.1. tag.fas (must have this name; fasta file with all tag sequences; optional).

5.1.1.2. adapter.fas (must have this name; fasta file with all adapters/vectors that might be present in the sequences; optional but strongly recommended).

5.1.1.3. fasta files from the sequencing project. There might be more than one file. The program deals with them one after the other. The name of the fasta files can have any alphanumerical characters and underscore and must have '.fas' extension (*e.g.* sample.fas). Everything in the definition line after '>' and before the first space is read as the sequence identifier. The identifier can have any alphanumerical characters and underscore. Replace all other characters by '\_'.



**5.1.2.** Open a terminal (START=>run=>cmd=>OK OR START =>Program =>Accessories => Command Prompt)



5.1.3. Change directory in a terminal to the folder that contains the scripts (e.g. `cd d:\QDD`)

5.1.4. Type `'perl pipe1.pl'`

```

C:\WINDOWS\system32\cmd.exe - perl pipe1.pl
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Emese Meglécz>d:
D:\>cd QDD
D:\QDD>perl pipe1.pl
*****
QDD version 13 November 2009
Emese Meglécz, Aix-Marseille University, Marseille, France
emese.meglecz@univ-provence.fr
Please, read the Documentation_QDD.pdf
*****

0: Operating system (win/linux): win
1: Input folder: data
2: Delete intermediate files (YES=1/NO=0): 1
3: Sort sequences by tag (YES=1/NO=0): 1
4: Remove adapter (YES=1/NO=0): 1
5: Minimum sequence length: 80
6: Minimum number of monobase repetitions in MS search: 10000000
7: Minimum number of dibase repetitions in MS search: 4
8: Minimum number of tribase repetitions in MS search: 4
9: Minimum number of tetrabase repetitions in MS search: 4
10: Minimum number of pentabase repetitions in MS search: 4
11: Minimum number of hexabase repetitions in MS search: 4
12: Pathway to BLAST: c:/BLAST2_2_18/bin/

Press enter if all of the settings are correct, or the number of the parameter i
f you wish to change the settings!

```

5.1.5. Follow the instructions on the screen to set the parameters and start the program

5.1.6. Once `pipe1.pl` is finished the main output files and the input file for `pipe2.pl` are found in 'pipe1\_xxx' subfolder of your input project folder.

5.1.9. Similarly, run `pipe2.pl` `pipe3.pl` and `pipe4.pl` by typing `'perl pipeX.pl'` (X= 2,3,4). The output files and the input files for the next step are found in `pipeX` subfolders of the project folder

## 6. Description of the outfiles

At the end of the run, there are four subfolders in your project folder.

Each of the subfolders pipe1\_XXX, pipe2\_XXX, pipe3\_XXX, pipe4\_XXX contain the most important outfiles of the step and the input files of the next step. (XXX is different in each run). All of these files are either fasta files or simple text files that can be opened by excel. The separators for the columns are semi-colons (;).

For each pipeX.pl a log file (pipeX\_log\_XXX.txt) is written with all input parameters (even the ones that are cannot be defined by the user), and summary statistics on the number of sequences and microsatellites.

If the 'delete intermediate files' are set to 0, intermediate files are kept. These are probably only of interest to the authors of QDD for troubleshooting and they are found in XXX subfolder of the QDD2 folder (XXX is a numerical identifier of the run).

QDD2 produces a number of temporary files, that all start by a numerical identifier of the run (e.g. 1308316501). If the run stops prematurely, these files might remain in the QDD2 folder. You can delete them without any loss of information.

### 6.1. Pipe1.pl

#### 6.1.1. sample\_NOTAG.tag (Produced only if tag=1)

Fasta file with sequences that did not have recognisable tags.

#### 6.1.2. sample.wov or (sample\_TAGY.wov if sequences were sorted by tag) . (Produced only if adapter=1)

Fasta file with adapters/vector/tag cut; 1 file for each tag; 'TAGY' is the tags name that are cut from the sequences.

e.g.

```
>FVU26NR06DF571_MID1
```

```
ACCATTGCTTTGACTGACAGATGAATTGACATTACATTTTCAGACAAAACAAAAAGCCCCACATTCGCTC  
TAAACACCCCTATCTGTCTCTGTCTCTGTGAAAACAGGCACATCCCACCTCAATAACAGATCAATCCC  
GCCGACATTTGGACATTTATTCAATTTTTCTCTCTCTCTCTCTCTCTCTCTGTCTCTGTTTCTCTTTT  
CCTACTCAAAGAATGAAAACGAAATTAACATTGAGCAAAAAGATAAATGGCGCCAACACGACAGCTCA  
AAACTCTCTCTGTTTATTGCTGAATG
```

Here the original sequence code is FVU26NR06DF571. The MID1 tag was removed from the sequence

#### 6.1.3. sample\_TAGY).woa. (Produced only if adapter=1)

Sequences that did not have adapter at the beginning (removed from further analyses)

#### 6.1.4. sample\_TAGY\_length.tbl . (Produced only if adapter=1)

text file with columns separated by ';'

Info on the number of bases cut from each sequence

Column1 Sequence code (e.g. FVU26NR06DF571\_MID1 )

Column2 Original length of the sequence (without tag) (e.g. 338 )

Column3 Number of bases cut from the beginning of the sequence (e.g. 18)

Column4 Number of bases cut from the end of the sequence (e.g. 20 )

Column5 Length of the sequence after cutting adapter/vector (e.g. 300 )

```
FVU26NR06DF571_MID1 ; 338 ; 18 ; 20 ; 300 ;
```

```
FVU26NR06DF6CK_MID1 ; 240 ; 24 ; 16 ; 200 ;
```

```
FVU26NR06DF6HF_MID1 ; 155 ; 24 ; 0 ; 131 ;
```

#### 6.1.5. sample\_TAGY\_80bp.seq

Text file with column separated by ';'; info on MS motif and position

Column1: Sequence code

Column2: number of microsatellites in the sequence

Column3: length of the sequence

Column4: motif of the first microsatellite

Column5: first position of the microsatellite Column6: last position of the microsatellite

Column7: number of repeats of the microsatellite

Columns4-7 are repeated for all microsatellites

e.g.

FVU26NR06DF571\_MID1\_A;2;300;TC;164;179;8;CT;278;285;4;

2 microsatellites were found, both with TC motif. Positions of the first microsatellites are 164-179 (inclusive) and 278-285 for the second. The numbers of repeats are 8 and 4, respectively.

#### **6.1.6. sample\_pipe2.fas**

Input file for pipe2; 1 file for each sample\_TAGY.wov; Only sequences that have microsatellites, and are longer than the user defined limit.

### **6.2. Pipe2.pl**

#### **6.2.1. sample\_pipe2\_consensus.fas**

Fasta file with all consensus sequences that did not have BLAST hit to grouped sequences  
Sequence code is a format of cons\_grX\_Y, where X is the identifier of a contig, and Y is the number of sequences in the contig.

If microsatellite polymorphism is detected the sequence identifier is followed by space and the microsatellite motif and its first and last position.

e.g. >cons\_gr12\_2 AC\_46\_55 (Contig 12 based on 2 sequences, a microsatellite with an AC motif from 46 to 55 position was found polymorph)

#### **6.2.2. sample\_pipe2\_gr.fas**

Fasta file with all grouped sequences (BLAST hit to other sequences but identity is low to include into a contig) The regions of the sequence covered by BLAST hits are printed in lower case.

#### **6.2.3. sample\_pipe2\_multihit\_css.fas**

Fasta file with sequences that had more than one hit to a sequence (possibly minisatellites)

#### **6.2.4. sample\_pipe2\_nohit\_css.fas**

Fasta file with sequences that did not have a BLAST autohit (sequence did not have a hit to itself, possibly Cryptically Simple Sequence)

#### **6.2.5. sample\_pipe2\_unique.fas**

Fasta file with all unique sequences (sequence that have only autohit in BLAST)

#### **6.2.6. sample\_pipe2\_cons\_subs.fas**

Fasta file with all sequences in contigs and their consensus sequences. Sequences of a same contigs are aligned. Pure microsatellites (from 5 repetitions) and homopolymers from 8 repetitions are printed in lower case.

#### **6.2.7. sample\_pipe3.fas (input file for pipe3.pl)**

All original unique sequences plus consensus sequences that did not have blast hit.

## 6.3. Pipe3.pl

### 6.3.1. sample\_pipe3\_primers.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	SEQUENCE_CODE	TARGET_REGION_FIRST_POS	TARGET_REGION_LENGTH_IN_BP	TARGET_MS_LENGTH_IN_REPEAT_NUMBER	PURE/COMPOUND	MOT_TRANS	MS_SEQ	POLYMORPH	DESIGN	BEST_PR_FOR_SEQ	BEST_PR_FOR_TARGET_REGION	PENALTY	PRIMER_LEFT_SEQUENCE	PRIMER_RIGHT_SEQUENCE	
8693	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		1	1.4.1562	210	AAACACTGACCCACAAAT		
8694	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.4.1908	212	AAACACTGATCCACAA		
8695	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.4.4550	211	AAACACTGATCCACAA		
8696	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.5.5413	123	AAACACTGAGGCTCTCT		
8697	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.5.9111	122	AAACACTGATGGCTCTCT		
8698	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.0119	295	ACTGCACACCGCTGTCT		
8699	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.1282	181	AAACACTGAGGTTTGTAT		
6700	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.3036	297	ACTGCACACACCGCTGT		
6701	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.3998	115	AAACACTGAATGAGG		
6702	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.4506	297	ACTGCACACACCGCTGT		
6703	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.4980	180	AAACACTGATGGTTTGTAT		
6704	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.4994	287	AAACACTGACCTCACTT		
6705	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.7191	164	AAACACTGATGCGAATT		
6706	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.6.8682	286	AAACACTGATCCTCACTT		
6707	cons_gr17_2	52	20	10	p	AC	ACACACAC/N	D		0	0.7.3105	288	AAACACTGACCTCACTT		
6708	cons_gr18_4	33	14	7	p	AC	ACACACAC/AC_33_46	F		0	1.7.7914	205	CAAACACAG/GCTTCACT		
6709	cons_gr18_4	33	14	7	p	AC	ACACACAC/AC_33_46	F		0	0.7.9354	204	AAACACAG/GCTTCACT		
6710	cons_gr18_4	33	14	7	p	AC	ACACACAC/AC_33_46	F		1	0.10.5563	146	AAACACAG/ATGTCTGTT		
6711	cons_gr18_4	33	14	7	p	AC	ACACACAC/AC_33_46	F		0	0.10.8655	148	AAACACAG/ATGTCTGTT		
6712	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		1	0.1.3103	126	AGCAGGAC/AGCGCTTC		
6713	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.0.6175	206	TCACTGGAC/AGCGCTTC		
6714	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.0.7504	129	TTTAGCAGG/AGCGCTTC		
6715	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.1.1436	127	TAGCAGG/AGCGCTTC		
6716	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.1.15947	202	GTGACAG/AGCGCTTC		
6717	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.1.19947	180	TGATAGAG/AGCGCTTC		
6718	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.2.5295	253	TGCTGTGTG/AGCGCTTC		
6719	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.2.5314	254	CTGCTGTG/AGCGCTTC		
6720	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.2.5455	162	CATGATAG/AGCGCTTC		
6721	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.2.6637	253	TGCTGTGTG/AGCGCTTC		
6722	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.2.7201	161	TGATAGAG/CAGCGCTT		
6723	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.4.3795	290	ACACCGTA/AGCGCTTC		
6724	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.4.3670	291	AACACCGTA/AGCGCTTC		
6725	cons_gr19_5	240	36	11	c	AC	CTCTCTCTCT/N	B		0	0.4.6184	290	ACACCGTA/AGCGCTTC		
6726	cons_gr1_2	34	18	6	p	AAAC	ACAACAAC/N	A		1	1.2.0479	139	GCTGCTCAATACTG		
6727	cons_gr1_2	34	18	6	p	AAAC	ACAACAAC/N	A		0	0.2.1857	140	CGCTGCTCAATACTG		
6728	cons_gr1_2	34	18	6	p	AAAC	ACAACAAC/N	A		0	0.2.1926	225	GCTGCTCAACACGAG		

Text file with columns separated by ‘;’

SEQUENCE\_CODE: original codes for unique and cons\_grX\_Y codes for consensus sequences

TARGET\_REGION\_FIRST\_POS: first position of target region in the sequence

TARGET\_REGION\_LENGTH\_IN\_BP: length of the target region in base pairs. If there is only one microsatellite is targeted (designs A-C and F), the target region covers the microsatellites (compound or pure). If more than one microsatellite is targeted (designs D, E, G), the target region includes the two most distant target microsatellites and the sequence between them.

TARGET\_MS\_LENGTH\_IN\_REPEAT\_NUMBER: length of the target microsatellite in repeat numbers. If microsatellite is compound, it is the number of repetition in the longest uninterrupted stretch. If there are more than one microsatellite in the target region, target MS info refers to the longest (in repeat numbers) of the target microsatellites.

PURE/COMPOUND: P=> Pure, C=>compound; . If there are more than one microsatellite in the target region, target MS info refers to the longest (in repeat numbers) of the target microsatellites.

MOT\_TRANS: AC, CA, TG, GT motifs are all indicated as AC motifs; If there are more than one microsatellite in the target region, target MS info refers to the longest (in repeat numbers) of the target microsatellites.

TARGET\_REGION\_SEQ: Sequence of the target region as found in the read

POLYMORPH: If polymorphism is detected, then the repeat motif is indicated.

DESIGN: A-G, see explanation at point 4.3.

BEST\_PR\_FOR\_SEQ: Among all primer pairs for the sequence select the lowest penalty for the best design type (A>B>C>...). Selecting lines with 1 in this column gives the total number of sequences with primers

BEST\_PR\_FOR\_TARGET\_REGION: Among all primer pairs for a target region select the lowest penalty for the best design type (A>B>C>...)>...). Selecting lines with 1 in this column gives the total number of target microsatellites with primers

PENALTY: Primer pair penalty (see documentation of Primer3)

PCR\_PRODUCT\_SIZE: in bp

PRIMER\_LEFT\_SEQUENCE:

PRIMER\_RIGHT\_SEQUENCE:

PRIMER\_LEFT\_DIST\_FROM\_MS: Distance between the target MS and the left primer in bp.



PRIMER\_RIGTH\_DIST\_FROM\_MS: Distance between the target MS and the righth primer in bp  
PRIMER\_LEFT\_FIRST\_POS: 5' end position of the left primer in the sequence  
PRIMER\_LEFT\_LENGTH: in bp  
PRIMER\_RIGHT\_FIRST\_POS: 5' end position of the right primer in the sequence  
PRIMER\_RIGHT\_LENGTH: in bp  
PRIMER\_LEFT\_TM: Annealing temperature of the left primer; see documentation of Primer3  
PRIMER\_RIGHT\_TM: Annealing temperature of the right primer; see documentation of Primer3  
PRIMER\_LEFT\_END\_STABILITY: see documentation of Primer3  
PRIMER\_RIGHT\_END\_STABILITY: see documentation of Primer3  
SEQUENCE: the whole sequence with homopolymers micro- and nanosatellites printed in lower case

6.3.2. sample\_pipe3\_targets.fas  
Fasta file with all sequences that have primers

## 6.4. Pipe4.pl

6.4.1. sample\_pipe4\_primers.csv  
Same file as sample\_pipe3\_targets.fas but includes columns on the NCBI blast hit information.  
Additional columns:  
Sequence\_code;  
Accession (best\_hit);  
Name (best\_hit);  
Description (best\_hit);  
E\_value (best\_hit);  
Score (best\_hit);  
TaxId (best\_hit)  
Superkingdom  
kingdom;  
phylum;  
class;  
family;  
genus;  
species

## 7. Troubleshooting

- 7.1. Pipeline 1 starts but the window closes immediately  
→ Do not run the perl script by clicking on the filename in explorer, but use the clean way of opening a terminal (5.1.2).
- 7.2. Pipeline 2 produces empty consensus alignments  
→ Make sure Clustal 2 is installed in the folder chosen at installation using the msi program.
- 7.3. Read access to some files is refused  
→ Make sure you run only one perl script at a time

## 8. References

Gilles, A., Meglècz, E., Pech, N., Ferreira, S., Malausa, T. and Martin, J F. 2011. Accuracy and quality assessment of 454 GS-FLX pyrosequencing. Submitted to **BMC Genomics**. 12:245.