

# QDD - version 3.1.2

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

**Windows and Linux  
Command line and Galaxy server version**

**[Emese Megléc](#) and [Jean-François Martin](#)**

Aix-Marseille Université, CNRS, IRD, UMR 7263 – IMBE, Equipe EGE Case 36, 3 Place Victor Hugo, 13331; Marseille Cedex 3, France

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

**[emese.meglec\(at\)imbe.fr](mailto:emese.meglec(at)imbe.fr)**

**<http://net.imbe.fr/~emeglec/qdd.html>**

**Disclaimer:** The software on this page is free to download and use, and thus comes with no warranty of any kind. While it hasn't caused us any problem, the current version of QDD is still considered as a beta version and you are responsible for any damages or loss of data you may sustain while using this software.

---

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#).

Correspondence, comments and bug reports about this program should be addressed to [Emese Megléc](#)

## Contents

QDD - version 3.....	1
A user-friendly program to select microsatellite markers and design primers from large sequencing projects.....	1
Windows and Linux Command line and Galaxy server version.....	1
Emese Megléc and Jean-François Martin.....	1
emese.meglec(at)imbe.fr.....	1
http://net.imbe.fr/~emeglec/qdd.html.....	1
Overview.....	3
Citation.....	4
General Description.....	4

PIPE1: Sequence preparation and microsatellite detection.....	4
PIPE2: Sequence similarity detection.....	4
PIPE3: Primer design.....	5
PIPE4: Contamination check and comparison to known transposable elements.....	5
Main output file(s): Primer table .....	5
How to chose primers from the primer table.....	6
Glossary.....	6
Download.....	8
QDD-VM.....	8
Download VM (same file for Linux and Windows):.....	8
QDD for manual installation.....	8
Download (same file for Linux and Windows):.....	8
QDD for installation into an existing Galaxy server.....	9
Easy update	
(Works both for the command line and VM).....	9
Update QDD-3.1 or QDD-3.1.1 to QDD-3.1.2.....	9
QDD documentation.....	9
Sample files.....	9
Earlier versions.....	10
Installation.....	11
Installing QDD-galaxy as Virtual Machine.....	11
Installing RepeatMasker Libraries (optional for running pipe4).....	16
Installing the NCBI nt database (optional for running pipe4).....	17
Installing QDD command line version on Linux.....	18
Installing QDD in command line version on Windows.....	19
Installing QDD into an existing local Galaxy server.....	21
Setting Parameters before the first run.....	21
Run QDD.....	23
Running QDD-Galaxy.....	23
Running QDD on command line.....	30
Running pipe1-4 separately .....	30
QDD.pl.....	30
Run all pipes in one go / batch submission / sorting sequences by tags.....	30
Examples for running QDD from the command line.....	31
Example1.....	31
Example2.....	32
Example3.....	33
Example4.....	33
List of parameters (Set in the set_qdd_default.ini file or on the command line).....	34
Output files.....	39
Naming output files in command line version.....	39
Output files of PIPE1.....	39
File names in Galaxy are in parentheses.....	39
Output files of PIPE2.....	40
File names in Galaxy are in parentheses.....	40
Output files of PIPE3.....	40
File names in Galaxy are in parentheses.....	40
Output files of PIPE4.....	42
File names in Galaxy are in parentheses.....	42
Trouble shooting.....	44

QDD VM does not boot.....	44
You followed the installation process described here but at point 6 the VM does not boot. .....	44
Primer3 error message: 'Unable to open file primer3_config.stack.ds'.....	44
QDD is running but you have error messages from primer3 and no primer is designed.....	44
PCR product size - primer position.....	44
PCR product size does not match with primer positions in the sequence.....	44

## Overview

In microsatellite development, high throughput sequencing has replaced the classical cloning based methods and in this process the first two versions of QDD played an important role by dealing with the essential bio-informatics steps leading from raw sequences to primer design.

The original version of QDD aimed to extract the best (putative) markers from a few Megabases (5-500 Mb) of 454 pyrosequences, since back in 2010 only this sequencing platform provided sufficiently long reads (300-500 bases). QDD1 treated all bioinformatics steps from raw sequences all the way to obtaining PCR primers: sorting sequences by tag, removing adapters/vectors, detection of microsatellites, detection of redundancy/possible mobile element association, and primer design.

QDD2 relaxed the primer design conditions, and let the users choose among more markers based mainly on the target region pattern.

QDD3 (current version) aims to improve primer design in the following ways

- The current length of Illumina sequencing (100-250 bp) is still slightly short for microsatellite marker development. However, sequencing of hundreds of Gbases of DNA has become affordable and thus – at least for not too big and repetitive genomes - sequences can be assembled into contigs. QDD3 does NOT do de novo assembly, but can take contigs (scaffolds/chormosomes) as an input, and extracts microsatellites with their flanking regions for primer design.
- Although the all-against-all comparison of the sequences in QDD versions 1 and 2 could pinpoint some of the putative interspersed or tandem repetitive regions, now a comparison to known repetitive elements via RepeatMasker is also available for the linux comand line version or in the virtual machine.
- In previous versions an automatic selection of the 'best' primer-pair was based arbitrary on the penalty score of the primer pairs calculated by [Primer3](#). However based on our wet lab results, this is not a good indicator of PCR success. The new choice of one primer pair per locus now depends on our wet lab tests, and provides more meaningful selection.
- Multi-threading for BLAST and RepeatMasker is possible.
- Fastq files can also be used as input file (the default option is the fasta format).


QDD3 is currently available in two different forms:

- A **command line version** that can be run both on linux and windows operating systems.
- QDD **integrated into a galaxy server** to provide an easy to use **interface** and many useful sequence treatment tools.

QDD scripts can be either downloaded directly, and they should be installed together with all the third party programs, or they can be downloaded pre-installed (as well as the third party programs) into a ready to use **virtual machine**. In both case, they can be run from Galaxy or from a command line.

[Back to Top](#)

## Citation

- 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](#). 
- Megléc, E., Pech, N., Gilles, A., Dubut, V., Hingamp, P., Trilles, A., Grenier, R. and Martin, JF. **2014**. QDD version 3.1: A user friendly computer program for microsatellite selection and primer design revisited: experimental validation of variables determining genotyping success rate. **Molecular Ecology Resources**, doi: 10.1111/1755-0998.12271. [Abstract](#)

[Back to Top](#)

## General Description

QDD is composed of four parts. Each of them can be run separately, or all at once in a pipeline

- **PIPE1: Sequence preparation and microsatellite detection**

The input sequences can be either assembled sequences (contigs, scaffold or chromosomes) or non-assembled sequences reads.

- **Assembled sequences** (contigs, scaffold or chromosomes): Microsatellites are extracted with a user defined flanking region on both sides. Adapter clipping is irrelevant in this case.
- **Non-assembled sequences**: Adapters can be removed (optional), short reads are eliminated and only sequences with microsatellites are kept for further analysis.

The input file format can be both fasta or fastq format. Fastq files are converted into fasta files without quality filtering. It is thus essential to appropriately trim low quality regions before starting QDD.

Input reads should be 100-1000 bp long, but QDD3 is not adapted for treating Gbases of short Illumina reads.

- **PIPE2: Sequence similarity detection**

Sequences are compared by an all-against-all BLAST and sorted according to their similarities into the following categories:

- **Singletons** => Singletons (the only BLAST hit is an autohit)
- **Nohit css** => Low complexity sequences (no BLAST hit to itself)
- **Multihit** => Putative minisatellites (more than one hit between a pair of sequences)
- **Grouped** => Sequences (including consensus) that had BLAST hit to other sequences, with below limit identity of the overlapping region. Regions covered by BLAST hits are masked.
- **Consensus** => All unique (no hit to grouped seqs) consensus sequences based on the alignment of reads, where the pairwise identity was at least 95% on the overlapping region. Consensus sequences are marked as polymorph if polymorphism in the microsatellite length is detected among the aligned reads.

**The main output file** (Input file for pipe3) contains all unique sequences (singletons and

consensus).

**If the sequences were extracted from assemblies**, making consensus sequences does not make sense. Pipe2 input sequences are still compared by an all-against-all BLAST, but only unique sequences (singletons) are kept for primer design.

- **PIPE3: Primer design**

Primers are designed in an iterative way for each sequence.

- By increasing size of the PCR products (To force design of primers that have longer products)
- From best to worst scenario for the target region (one single perfect microsatellite => multiple microsatellites, homopolymers, nanosatellites)

- **PIPE4: Contamination check and comparison to known transposable elements**

- **Contamination check:** An optional contamination check can be done by BLASTing all sequences with successful primer design against the nt database of NCBI and checking the taxonomic classification of the best hit. This step can be done either by a local BLAST or by a remote BLAST. The first option requires the download of the nt database from NCBI (ca.15 Gb) and it is faster than the second. The second option relies on a good internet connection, it is much slower and connection time outs can be frequent.

This step does not pick out particular sequences as contaminant, but can warn the users of serious general contamination (or mixing up samples) if taxonomic groups of the best hits do not match the target species.

- **Comparison to known transposable elements:** Sequences with successful primer design can be compared to known transposable elements by running RepeatMasker from QDD (available in the VM version or in command line version run on linux system)

[Back to Top](#)

## Main output file(s): Primer table

The main output files are the primer tables produced by pipe3 and completed by pipe4. They are named **xxx\_pipe[3,4]\_primers.tabular** in command line version and **Table with primers** in Galaxy.

Each line corresponds to a primer pair and there are several primer pairs designed for each sequences.

For each primer pair information is given on

- the sequence
- the target region (number, type, motif, length of microsatellite),
- the primers (position length, annealing temperature...).

Pipe4 completes the output table of pipe3 with

- information on the best hit against Genbank (accession, e-value, score, taxonomy) and
- best hit to a known transposable element.

Take your time to understand the information in the different columns in this table, since it helps you to choose markers and primer pairs out of the many designed. (see detailed description of the columns in [Output files](#) section)

[Back to Top](#)

## How to choose primers from the primer table.

The following suggestions are based on our lab tests (Megléc et al submitted) or simply on common sense. The title of the column in the output file relevant for each selection criterion is indicated in capital letters.

- **ONE\_PRIMER\_FOR\_EACH\_SEQ**: An automatic selection of 1 primer pair for each locus, based on our lab tests. (See detailed description of the columns in [Output files](#) section)
- Avoid primers with high alignment score to the sequence (**PCR\_PRIMER\_ALIGNSCORE**, annealing sites are not considered for calculation of the alignment score)
- Better to have a pure microsatellite, then a compound (**PURE/COMPOUND**)
- Microsatellites with more repeats are more likely to be polymorphic (**TARGET\_MS\_LENGTH\_IN\_REPEAT\_NUMBER**)
- Avoid microsatellite motifs that can form hairpin (e.g. (AT)<sub>n</sub>; **MOT\_TRANS**)
- Choose markers in different ranges for PCR product length to facilitate multiplexing (**PCR\_PRODUCT\_SIZE**)
- Avoid primers that are very close (>20 bp) to the target microsatellite (**MIN\_PRIMER\_TARGET\_DIST**)
- Choose compatible annealing temperatures of the primers if you have changed the default values of primer3. According to the default values, all primer TM could vary between 57 and 63 C, which makes most primers pairs compatible for multiplexing (**PRIMER\_LEFT\_TM**, **PRIMER\_RIGHT\_TM**)
- If input sequences were contigs, avoid selecting markers that are near to each other on the same contig (**FIRST\_POS\_ON\_CONTIG**, **CONTIG\_CODE**)
- If you have run RepeatMasker, avoid primers with good hits to transportable elements (**RM\_score**; High score indicates a good alignment between TE and your sequence)
- Consensus sequences based on read numbers much higher than expected coverage are probably derived from different loci of a repetitive element, thus should be avoided (**NUMBER\_OF\_READS**).
- Prefer target regions that do not have multiple microsatellites, nanosatellite, homopolymers (**DESIGN A, B**)

[Back to Top](#)

## 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.

[Back to Top](#)

---

# Download

## QDD-VM

QDD-VM (QDD in a virtual machine) is an easy but greedy installation. The virtual machine (VM) comes with QDD and (almost) all its dependencies installed in a Galaxy server. The installation is easy and you can either run QDD from a terminal (command line) or from your local Galaxy server installed into the VM, which provides a user friendly interface. However, the down-loadable file is very large (2.2 Gb) and you will need at least 1 GB of RAM and 50 Gb of storage for the VM. The 50 Gb of storage is dynamically allocated, which means this is the maximal volume used on the host machine, but the actual storage size depends on the quantity of data you keep on the VM. A collateral bonus is that after installing the VM version of QDD, you will also benefit from a local general purpose galaxy server and you will be able to use it for NGS data manipulation (<http://galaxyproject.org/>, <http://wiki.galaxyproject.org/>).

If you are not familiar with Virtual machines, the user manual of Oracle VM VirtualBox gives a very good introduction, or plenty of details, in case you want to know more. (<https://www.virtualbox.org/manual/UserManual.html>).

**Download VM (same file for Linux and Windows):**

**QDD-VM version 3.1.2 ( 2.1 Gb ; 04 July 2014)**

- If you have QDD3.1 or QDD3.1.1 already installed see [Easy update](#)

**QDD-VM version 3.1.1 ( 2.1 Gb ; 17 Apr 2014)**

- If you have QDD3.1 already installed see [Easy update](#)

**QDD-VM version 3.1 ( 2.2 Gb ; 28 Jan 2014)**

**QDD-VM version 3.0 (beta) ( 3.4 Gb ; 04 Oct 2014)**

## QDD for manual installation

QDD is written in Perl and runs as a standalone application on Windows or Linux systems from a terminal (command line version).

This is a light and flexible version of QDD that can do the same analyses as the Galaxy version.

You can even change more parameters than with Galaxy and use batch submission if you have many files to run. The downside is that there is no graphical interface and you need to [install all the dependencies of QDD](#).

**Download (same file for Linux and Windows):**

**QDD version 3.1.2 ( 127 Mb; 04 July 2014 )**

- If you have QDD3.1 or QDD3.1.1 already installed see [Easy update](#)

**QDD version 3.1.1 ( 127 Mb; 17 Apr 2014 )**

- If you have QDD3.1 already installed see [Easy update](#)

**QDD version 3.1 ( 106 Mb; 28 Jan 2014 )**



[QDD version 3.0 \(beta\)](#) ( 106 Mb; 04 Oct 2014 )

## QDD for installation into an existing Galaxy server

Same files as for [QDD for manual installation](#)

### Easy update

**(Works both for the command line and VM)**

#### Update QDD-3.1 or QDD-3.1.1 to QDD-3.1.2

- Download the update file : [update\\_QDD-3.1\\_to\\_QDD-3.1.2.tar.gz](#) and extract the files.
- You will find 8 files in the extracted folder.
- Replace the original files in the folder that contains QDD-3.1 or QDD-3.1.1 scripts (in the VM it is ~/galaxy\_dist/tools/qdd) by these new files.
- It is ready. All the settings that you have chosen in the set\_qdd\_default.in are still valid

### QDD documentation

[QDD documentation pdf](#) ( 718 kb )

### Sample files

All sample files are archives of input files and the output files produced by QDD-3.1.2 with the commads specified bellow.

- [example1.tag.gz](#) ( 50 kb) **Contig** from an insect  
perl QDD.pl -input\_folder c:\data\_example1 -contig 1 -make\_cons 0 -rm 1 -rm\_lib insecta  
See [example1](#) at running QDD for more detail
- [example2.tag.gz](#) ( 103 kb) **454 reads without adapters and tags**  
perl QDD.pl -input\_folder c:\data\_example2 -check\_contamination 1  
See [example2](#) at running QDD for more detail
- [example3.tag.gz](#) ( 13 kb) **454 reads with adapters and tags**  
perl QDD.pl -input\_folder c:\data\_example3 -tag 1 -tag\_file c:\myfolder\tag.fas -adapter 1  
-adapter\_file c:\myfolder\adapter.fas -check\_contamination 1  
See [example3](#) at running QDD for more detail
- [example4.tag.gz](#) ( 330 kb) **Illumina reads in fastq format without adapters and tags**  
perl QDD.pl -input\_folder c:\data\_example4 -fastq 1 -check\_contamination 1 -rm 1 -rm\_lib  
vertebrates  
See [example4](#) at running QDD for more detail

## **Earlier versions**

# Installation

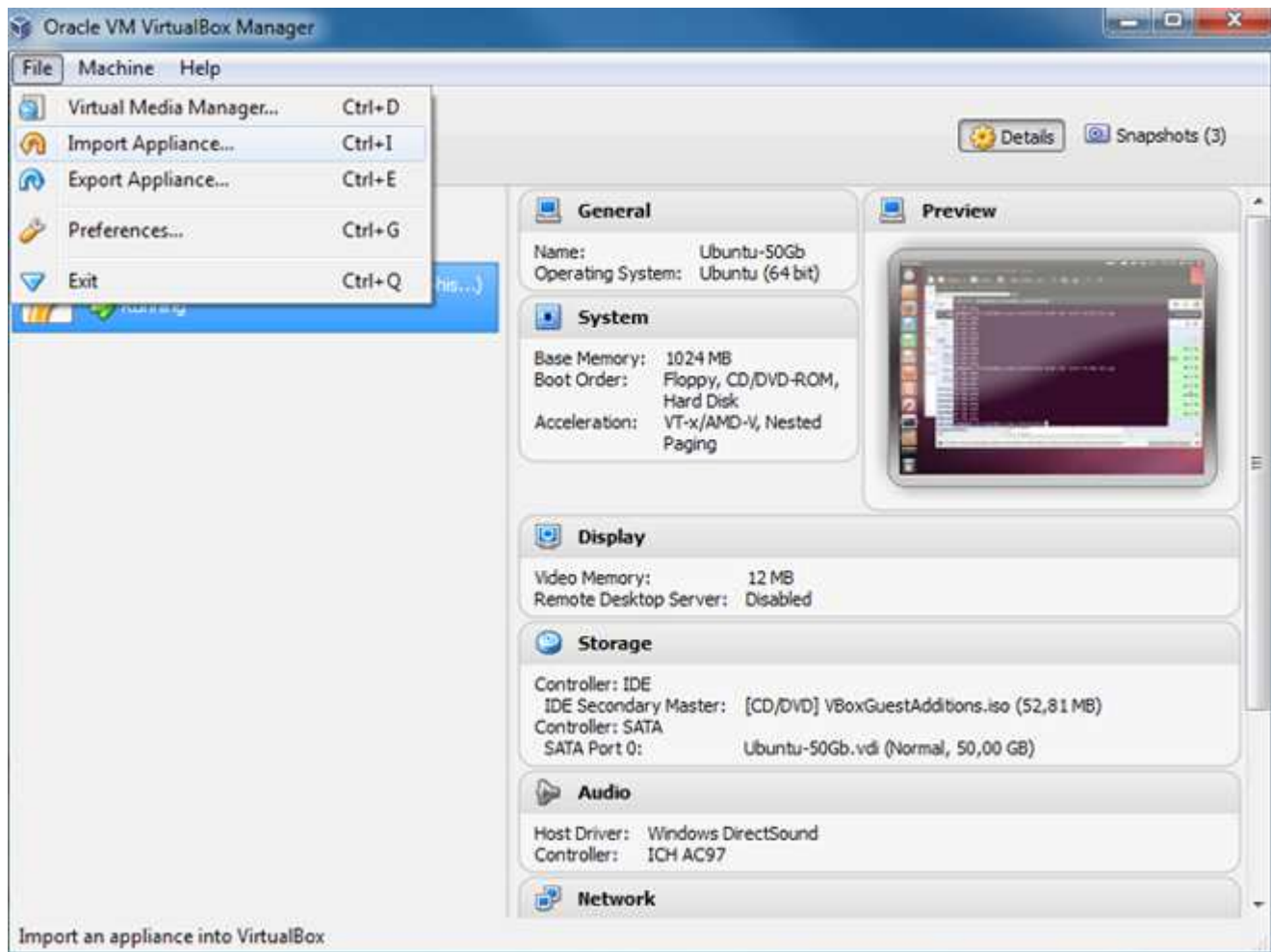
## Installing QDD-galaxy as Virtual Machine

1. Download and install Oracle virtual box appropriate to your computer from <https://www.virtualbox.org/wiki/Downloads>.

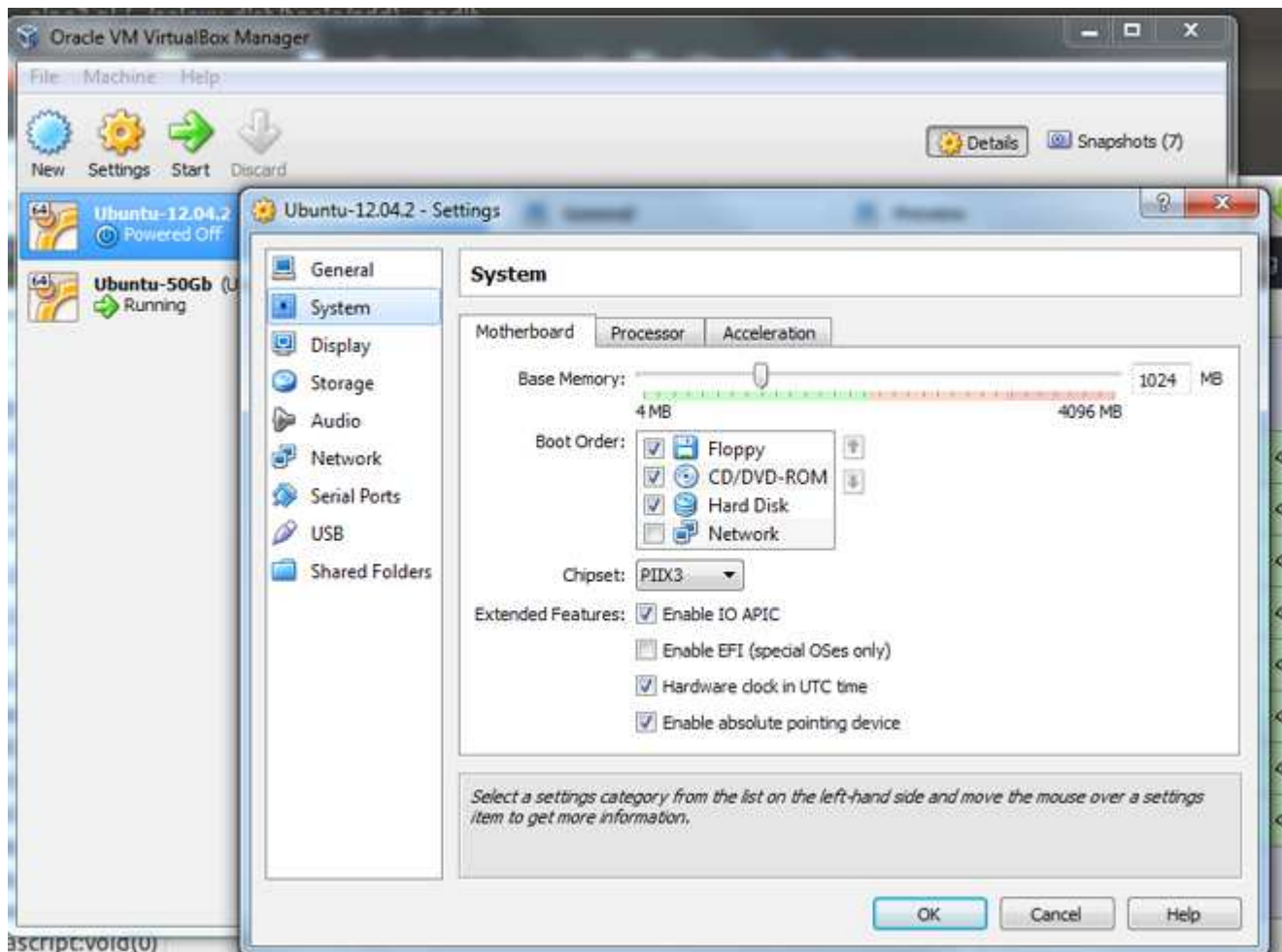
The virtual Box is a manager that can handle different Virtual Machines.

- Host system refers to the Operating system of your computer.
  - The guest operating system (the operating system of the VM) is Linux (Ubuntu-12.04.2).
  - VirtualBox comes in many different packages, and installation depends on your host operating system. If you have installed software before, installation should be straightforward: on each host platform, VirtualBox uses the installation method that is most common and easy to use. If you run into trouble or have special requirements, please refer to [Chapter 2: Installation details of the manual of Virtual Box](#) for details about the various installation methods.
  - You do not need to install extension packs.
2. Download the Virtual Machine with QDD from the [download page](#)
  3. Start the Virtual Box (see [manual of Virtual Box](#) for help)
  4. Import the VM into the Virtual Box by choosing the 'Import appliance' in the 'File' menu of the Virtual Box. (for more details see [manual of Virtual Box](#) ).

When importing the VM, some settings - like the memory allocated to the VM and the number of CPUs - can be directly modified in the import window. By default 1 GB of RAM, and 1 CPU is set. You can also adjust these parameters later (see point 5).



5. **Optional:** By default 1 GB of RAM, and 1 CPU is set for the VM you have just imported. However, you can adjust the RAM and the number of CPU if you have a higher capacity. (<https://www.virtualbox.org/manual/ch01.html#configbasics>)



- Open a the 'Settings' window by clicking on the Settings button in the menu.
- Select 'System' in the left panel of the window.
- Select the 'Motherboard' tab for changing the Base Memory (RAM). This sets the amount of RAM that is allocated to the VM when it is running. The specified amount of memory will be requested from the host operating system, so it must be available or made available as free memory on the host when attempting to start the VM and will not be available to the host while the VM is running. (or details see [www.virtualbox.org/manual/ch03.html#settings-motherboard](http://www.virtualbox.org/manual/ch03.html#settings-motherboard) )
- Select the 'Processor' tab for changing the number of CPU. You should not, configure virtual machines to use more CPU cores than you have available physically. ( for details see <https://www.virtualbox.org/manual/ch03.html#settings-processor>)

**Beware!** If you have changed the number of CPU available to your VM, you will have to change the `-num_threads` parameter in the `set_qdd_default.ini` file See [Setting parameters](#))

#### 6. Start the Virtual Machine

Double-click on its entry in the list within the Manager window. This opens up a new window, and the virtual machine which you selected will boot up. Everything which would normally be seen on the virtual system's monitor is shown in the window. In general, you can use the virtual machine much like you would use a real computer.

Check the [trouble shooting](#) page if you have a problem at this stage.

#### 7. Keyboard setting

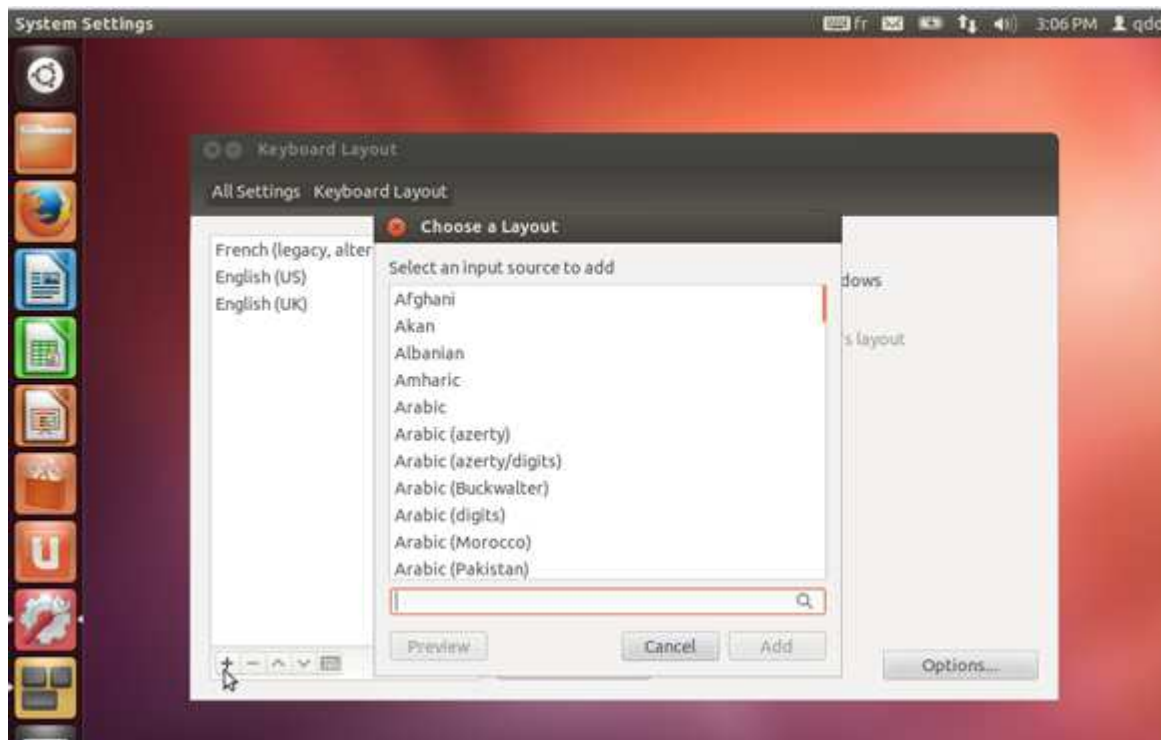
- Choose the keyboard setting closest to the one you are using, by clicking on the keyboard icon on the top right corner of your VM screen.



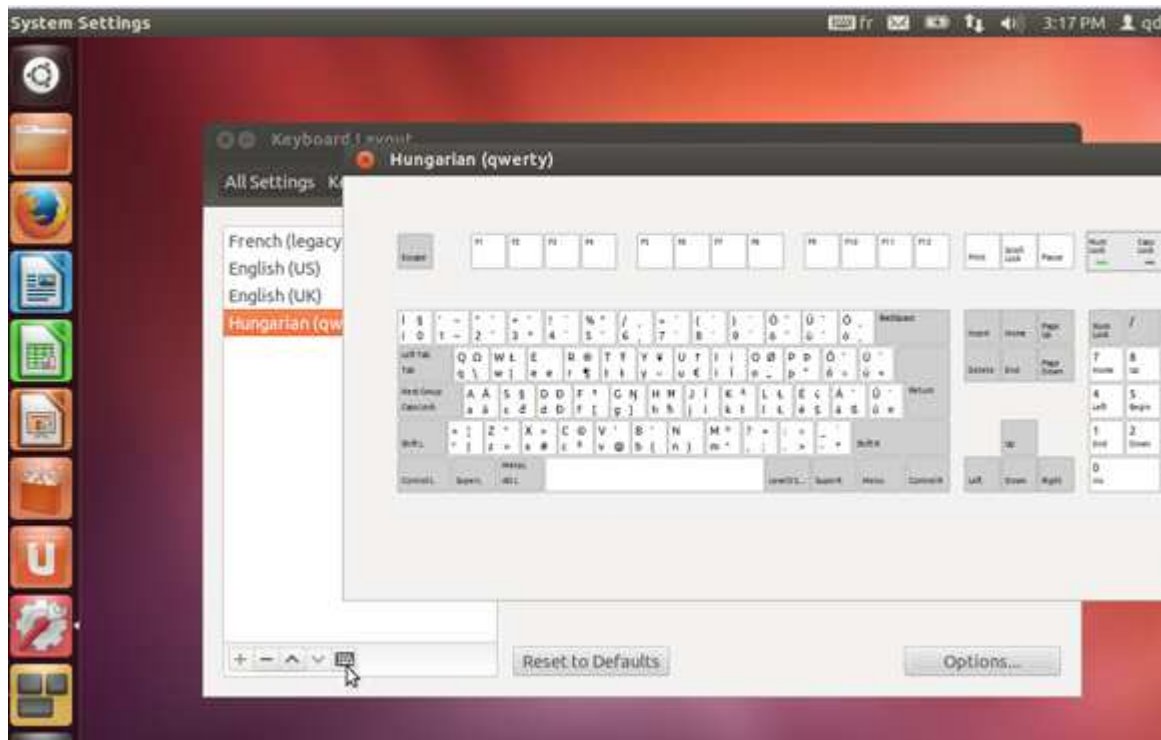
- Type the password 'qddGalaxy', to access the machine.
- If your keyboard does not correspond to the any of three proposed (English UK, English US, French) you can add yours.
  1. Click on the keyboard icon on the on the top right corner of your VM screen, and select 'Keyboard Layout Settings...'



2. Click on the '+' icon on the bottom left of the 'Keyboard Layout' window, to see available keyboard settings.



3. Choose the keyboard of your choice and click on 'Add'
4. This keyboard will appear in the left panel. Select it and click on the keyboard icon at the bottom left, to check if it corresponds to yours.



5. Use the arrow icons at the bottom left of the window, to put this keyboard at the top of the list.



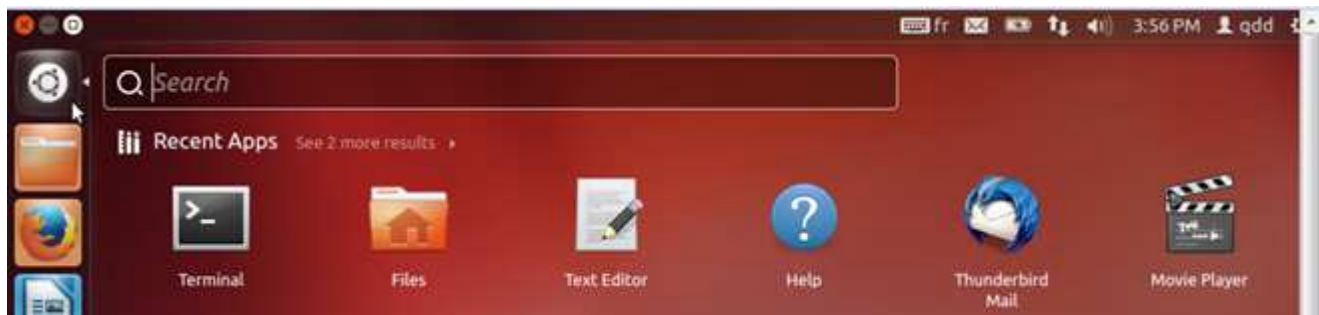
6. Close the window, it's all set....

...Well almost all set. In the VM you have just installed, QDD, Galaxy server and all essential third party programs are installed and ready to use. However, you cannot use RepeatMasker without downloading the RepeatMasker Libraries from GIRI ([Genetic Information Research Institute](http://www.girinst.org)). Since this database of repetitive elements is available freely only to academic users, we could not include it into the VM.

The other database you might need is the nucleotide database of NCBI. It is about 15 Gb and regularly updated, therefore it is better to download the last version directly to your VM.

### Installing RepeatMasker Libraries (optional for running [pipe4](#))

1. Start a browser (Firefox) in the left application list of the VM display. Register at [GIRI](http://www.girinst.org)
2. Download repeatmaskerlibraries-[version].tar.gz from <http://www.girinst.org/server/RepBase/index.php>
3. Open a terminal by clicking on the Ubuntu icon in the top left corner of the VM display and type 'terminal' in the search box, and then click on the terminal icon that appears



4. In the terminal type (or copy) the following commands which will copy the repeatmasker libraries into the RepeatMasker folder:

```
sudo cp ~/Downloads/repeatmaskerlibraries* /usr/local/RepeatMasker/
```

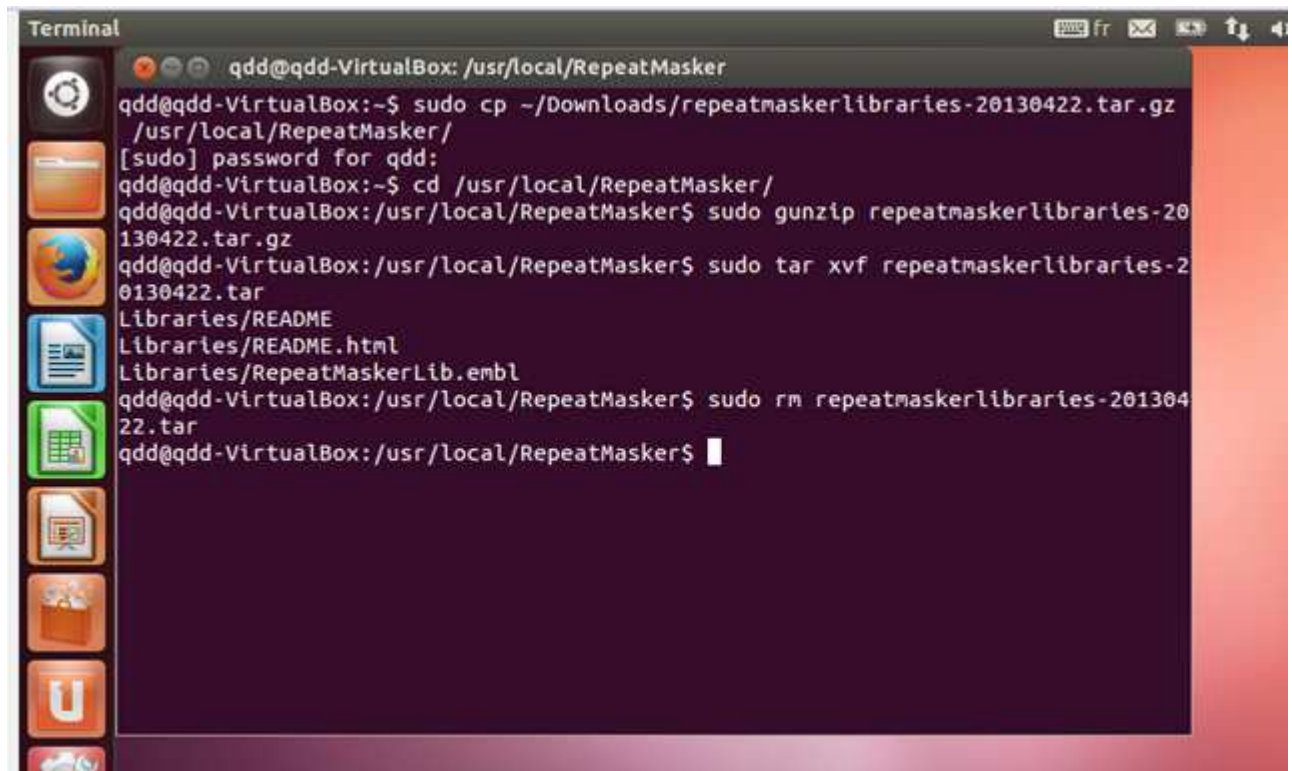
(sudo is a command that allows you to modify protected files. It will prompt you to type the



password 'qddGalaxy'. Don't be surprised if the cursor is not moving when you type the password.)

Now type (or copy) the following commands to unpack the repeatmasker library:

```
cd /usr/local/RepeatMasker
sudo gunzip repeatmaskerlibraries-*.tar.gz
sudo tar xvf repeatmaskerlibraries-*.tar
sudo rm repeatmaskerlibraries-*.tar
```

A terminal window titled 'Terminal' showing the installation of RepeatMasker libraries. The user is logged in as 'qdd' on a 'qdd-VirtualBox' machine. The terminal shows the following commands and output:

```
qdd@qdd-VirtualBox: /usr/local/RepeatMasker
qdd@qdd-VirtualBox:~$ sudo cp ~/Downloads/repeatmaskerlibraries-20130422.tar.gz /usr/local/RepeatMasker/
[sudo] password for qdd:
qdd@qdd-VirtualBox:~$ cd /usr/local/RepeatMasker/
qdd@qdd-VirtualBox:/usr/local/RepeatMasker$ sudo gunzip repeatmaskerlibraries-20130422.tar.gz
qdd@qdd-VirtualBox:/usr/local/RepeatMasker$ sudo tar xvf repeatmaskerlibraries-20130422.tar
Libraries/README
Libraries/README.html
Libraries/RepeatMaskerLib.embl
qdd@qdd-VirtualBox:/usr/local/RepeatMasker$ sudo rm repeatmaskerlibraries-20130422.tar
qdd@qdd-VirtualBox:/usr/local/RepeatMasker$
```

#### Installing the NCBI nt database (optional for running [pipe4](#))

1. [Open a terminal](#) (as for installing RepeatMasker) and make a folder /usr/local/nt/ and change folder by typing (or copying) the following commands:

```
sudo mkdir /usr/local/nt/
cd /usr/local/nt/
```

2. Download the database:

```
sudo wget ftp://ftp.ncbi.nlm.nih.gov/blast/db/nt*
```

This step is long (>15GB). It is better to download the files overnight.

3. Check the integrity of the downloaded files:

```
sudo md5sum --check nt.*.tar.gz.md5
```

You should see OK for each of the xxx.tar.gz files

4. Unpack the files:

```
sudo tar -zxvf nt.00.tar.gz
```

Repeat this operation to all nt.##.tar.gz files

5. Remove packed files and md5 files:

```
sudo rm nt.*.tar.gz*
```

[Back to Top](#)

## Installing QDD command line version on Linux

1. Extract QDD files to a folder (if installing QDD into an existing galaxy server, this folder should be ../galaxy-dist/tools/qdd. For command lines below I will use /usr/local/qdd)

```
mkdir /usr/local/qdd
cd /usr/local/qdd
tar -xvfz qdd.tar.gz
```

2. Install Perl, blast+, clustalW2, Primer3, RepeatMasker (optional), bioperl (optional), NCBI nt database (optional)
  - **Perl** (<http://www.perl.org/get.html>). It is likely that Perl is already installed. To check it type 'perl -v' on the command line.
  - **Bioperl** (<http://www.bioperl.org/>); It is only necessary for contamination check
  - **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/>) You can use either the more recent version 2 of Primer3 or an older Primer3-1.1.4 version.
  - **RepeatMasker** (<http://www.repeatmasker.org/>) Optional for checking similarities to know transposable elements. When using RepeatMasker you also need a library of repetitive elements from GIRI (<http://www.girinst.org/>). It is free only for academic users.
  - Download **NCBI nt database** (<ftp://ftp.ncbi.nlm.nih.gov/blast/db/>). For help see '[Installing NCBI nt database](#)'. For contamination check you can either use a remote BLAST or a local BLAST. Only for the local BLAST you need to download this database, but it is recommended, if you are likely to have many sequences to test. (see [Contamination check](#))
3. Make a symbolic link to subprogramQDD.pm and ncbi\_taxonomy.pm in /etc/perl/subprogramQDD.pm

```
ln -s /usr/local/qdd/subprogramQDD.pm /etc/perl/subprogramQDD.pm
ln -s /usr/local/qdd/ncbi_taxonomy.pm /etc/perl/ncbi_taxonomy.pm
```
4. Make a hard link to set\_qdd\_default.ini in /etc/qdd/ (Beware! If you are re-installing qdd, remove the old link in /etc/qdd/ and make a new)

```
mkdir /etc/qdd/
ln /usr/local/qdd/set_qdd_default.ini /etc/qdd/set_qdd_default.ini
```
5. Edit the **set\_qdd\_default.ini** file to set parameters that are constant (see details in [Setting Parameters](#))
  - Make sure the operating system is set to linux (syst=linux)
  - Make sure galaxy is set to 0
  - Set the full path to BLAST+, ClustalW, Primer3 and RepeatMasker executables (blast\_path, clustal\_path, primer3\_path, rm\_path). This step is necessary for Primer3 but you can ignore it for the other three software if they are in your PATH.
  - Set the version of primer3 (primer3\_version)

- Set the full path to the QDD scripts (qdd\_folder) and the output folder that will contain temporary files (out\_folder). This folder must exist before running QDD.
- If you are using local blast for contamination check, set local\_blast to 1 and set the name and path of the downloaded ncbi database (blastdb)
- Set the number of threads (number of CPU) in num\_threads (used for BLAST and RepeatMasker).

[Back to Top](#)

## Installing QDD in command line version on Windows

1. Untar and unzip [QDD\[version\].tar.gz](#) into a folder where you would like to install it

For windows, you can use WinRar (<http://freedownloadwinrar.org/>) for decompressing the file. Keep all the files in one folder (referred to as qdd\_folder in the parameters). Choose a place for your qdd\_folder with no space in the path, and avoid using file names with spaces in it.

2. Install Perl, blast+, clustalW2, Primer3, bioperl (optional), NCBI nt database (optional)
  - **ActivePerl** ( <http://www.activestate.com/activeperl/> )
  - **Bioperl** ( <http://www.bioperl.org/> ; It is only necessary for contamination check. Help at [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.  
Install ClustalW2 using the msi file and keep the files within the folder selected during the installation process.
  - **Primer3** ( <http://primer3.sourceforge.net/>) You can use either the more recent version 2 of Primer3 or an older Primer3-1.1.4 version, which is easier to install in windows.
  - Download **NCBI nt database** ( <ftp://ftp.ncbi.nlm.nih.gov/blast/db/>). For contamination check you can either use a remote BLAST or a local BLAST. Only for the local BLAST you need to download this database, but it is recommended, if you are likely to have many sequences to test. (see [Contamination check](#))
    - Open a terminal. (Program =>Accessories => Command Prompt)



- In the terminal type:
 

```
perl [blast_path]update_blastdb.pl nt
```
  - Decompress the downloaded files. You can use WinRAR (<http://freedownloadwinrar.org/>) for decompressing the files under windows.
3. Edit the **set\_qdd\_default.ini** file to set parameters that are constant (see details in [Setting Parameters](#))
- Make sure the operating system is set to windows (syst=win)
  - Make sure galaxy is set to 0
  - Set the full path to above installed executables after the appropriate parameter names (blast\_path, clustal\_path, primer3\_path)
  - Set the version of primer3 (primer3\_version)
  - Set the full path to the QDD scripts (qdd\_folder) and the output folder that will contain temporary files (out\_folder). This folder must exist before running QDD.
  - If you are using local blast for contamination check, set local\_blast to 1 and set the name and path of the downloaded ncbi database (blastdb)
  - Set the number of threads (number of CPU) in num\_threads (used for BLAST).

[Back to Top](#)

## Installing QDD into an existing local Galaxy server

1. Modify the `~/galaxy-dist/tool_conf.xml` file to include QDD tool xml files.

After add:

```
<section name="QDD" id="QDD3">
  <tool file="qdd/QDD_pipe1.xml" />
  <tool file="qdd/QDD_pipe2.xml" />
  <tool file="qdd/QDD_pipe3.xml" />
  <tool file="qdd/QDD_pipe4.xml" />
</section>
```

2. Follow the same steps as [Installing QDD in command line version for Linux](#). Except for setting galaxy to 1 in the `set_qdd_defaults.ini` file.

[Back to Top](#)

## Setting Parameters before the first run

The default parameters of QDD are stored in `set_qdd_default.ini` file. You should edit this file for changing default settings that depend on your computer configurations and file locations (e.g. path to executables, Primer3 version, directory for output files).

Lines starting by `#` are comments to give you the meaning of the parameters in the next line(s)

All other lines set one parameter at a time. The name of the parameter is followed by '=' then by the value of the parameter. Do NOT change the name of the parameter or delete the equal sign.

If the executables of BLAST+, CLUSATW2, Primer3 and RepeatMasker are not in your \$PATH, give the full path to these software after the appropriate parameter name. Otherwise you can leave these values empty.

Although you can set all parameters in the command line at every run, the following parameters are likely to be stable for all runs, thus better to be edited in the `set_qdd_default.ini` file:

```
# run QDD from galaxy server; 1 for yes 0 for running QDD from terminal
galaxy=0

#operating system [linux/win]
syst =

# Full path to blast executables (including the bin folder) e.g. C:\Program Files\NCBI\blast-
2.2.25+\bin or /home/EM/blast/bin) If the folder is in your path it can be left empty.
blast_path=

# Full path to clustalw executables. If the folder is in your path it can be left empty.
clust_path =

# Full path to primer3_core executable. (e.g. C:\primer3-1.1.4-WINXP\bin)
primer3_path =

# Primer3 version [1/2]
primer3_version =

# Full path to qdd scripts
qdd_folder =

# Output folder name with full path. Must exist before run. If not specified output files are
written in the current working directory
```

```
out_folder =
#[0/1] (1 for deleting temporary files after the run)
del_files = 1
#[0/1] (1 for printing out supplementary information, only needed for debugging)
debug = 0
# name (including full path) to a local database (nt of NCBI; e.g. /usr/blastdb/nt); Only
needed if local BLAST is used for contamination check
blastdb = d:\blastdb\nt
#number of threads for BLAST and RepeatMasker (number of CPU)
num_threads = 1
# [0/1] 1:run local blast for contamination check; 0:run remote blast for contamination check
local_blast = 0

#PIPE4 SPECIFIC PARAMETERS
# Full path to RepeatMasker executables. (e.g. /usr/local/RepeatMasker/) If the folder is in
your path it can be left empty.
rm_path = /usr/local/RepeatMasker/
```

# Run QDD

## Running QDD-Galaxy

1. Start up the galaxy server.

- Open a terminal (by clicking on the Ubuntu icon in the top left corner of the VM display and type 'terminal' in the [search box](#))
- Type in the terminal:

```
cd ~/galaxy-dist/  
sudo sh run.sh
```

You will be prompted to type the password (**qddGalaxy**).

You will see plenty of messages on the screen.

Wait till you see

```
-serving on http://127.0.0.1:8080
```

- Leave the terminal open. You can close it by typing **Ctrl C** only when you have finished with the Galaxy server.
2. Connect to the local galaxy server from your web browser at <http://127.0.0.1:8080/>
3. When Galaxy starts you will find 3 panels
- The left shows you the different tools available
  - The right shows the files in your history
  - The middle contains different information according to the context (help and input settings for the tools, contents of the output files)

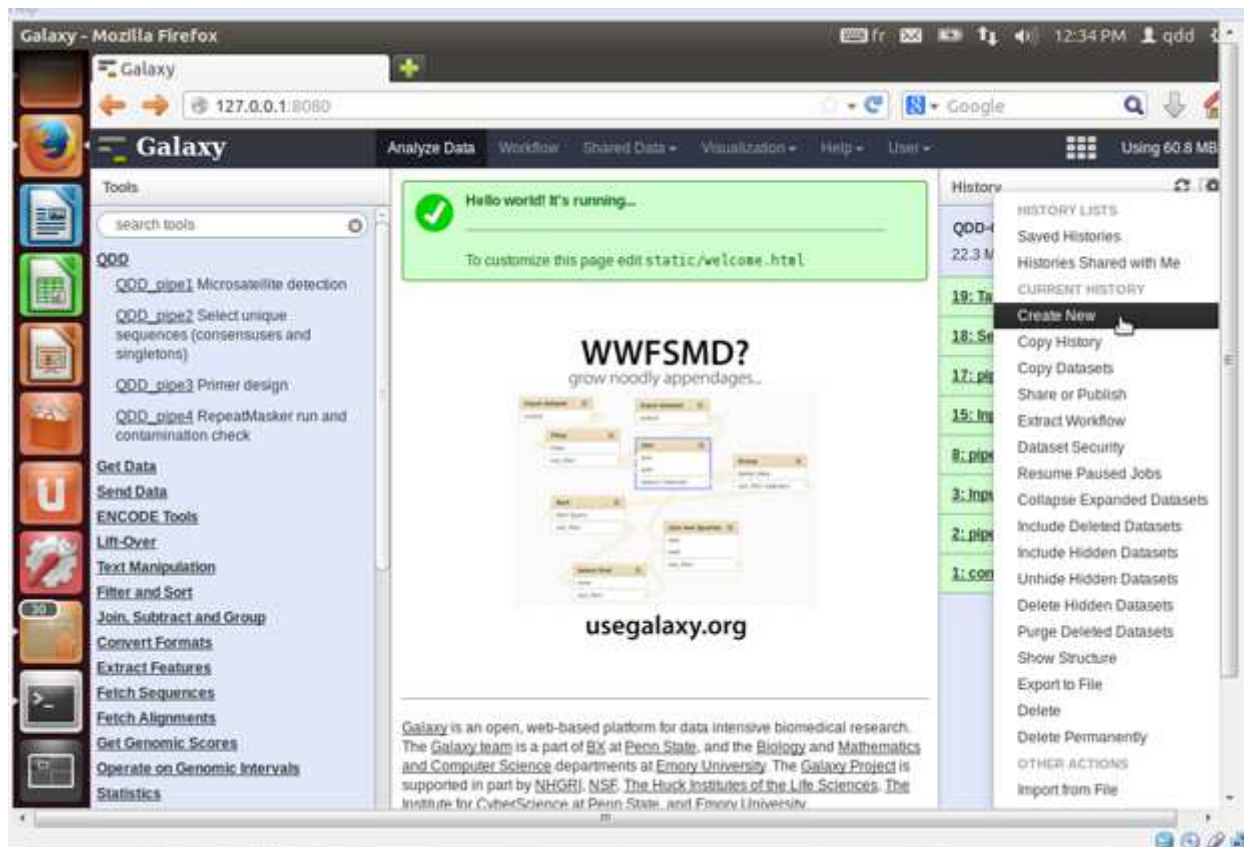


4. Create an account in the User menu (black line on top of the page), so you can save your

histories, workflows, share you data etc.

If you are using Galaxy in the VM there is already an account you can use. It already contains qdd workflows and sample histories:

- **Email:** qddGalaxy@gmail.com
  - **Pwb:** qddGalaxy
  - **public name:** qdd-galaxy
5. First you need to send your data files to Galaxy.
- Create a new history by clicking on the dented wheel on the top right, and selecting 'Create New'.

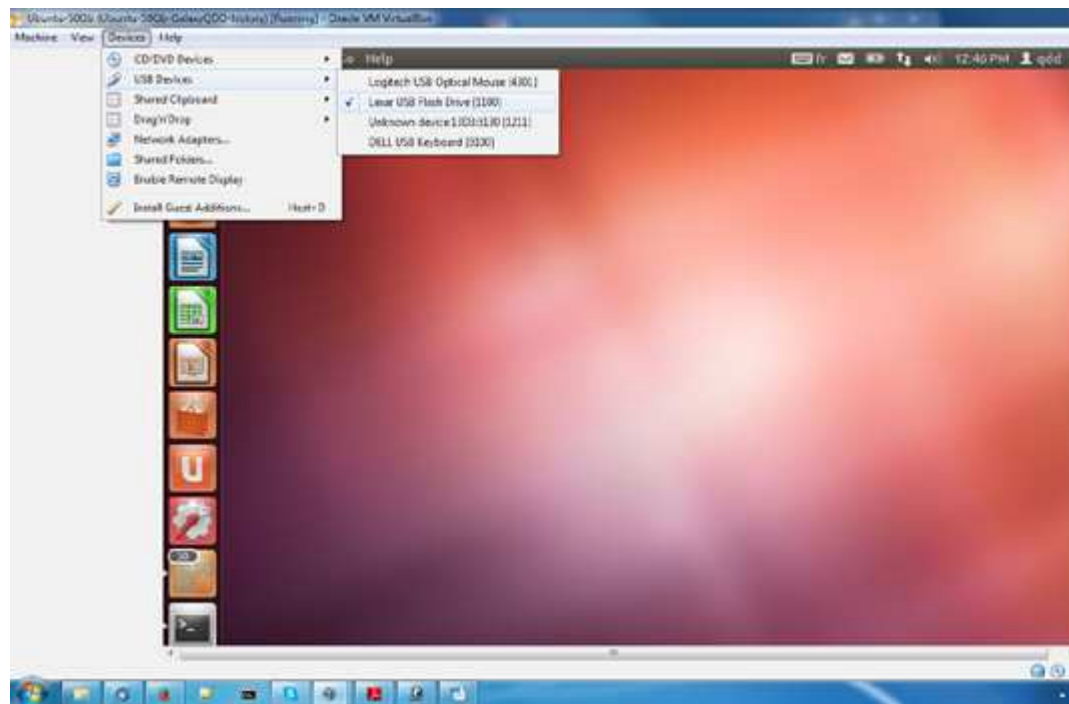


- User => Saved histories => Rename  
You can rename your history by selecting 'Saved histories' in the 'User' menu, and selecting and renaming your current history.
- Get Data => Upload file  
Select 'Upload file' from the 'Get Data' menu in the left panel.  
You can either use the browser to find the file in your computer, or copy the URL from which it can be uploaded.  
When first using Galaxy it is better to use the example files found in **/home/qdd/galaxy-dist/tools/qdd/data** of the VM.  
To get your own files to the VM, you can either
  - Set up a **shared folder** between the host system and the guest system (see documentation at [www.virtualbox.org](http://www.virtualbox.org)) or
  - use an **external drive**.

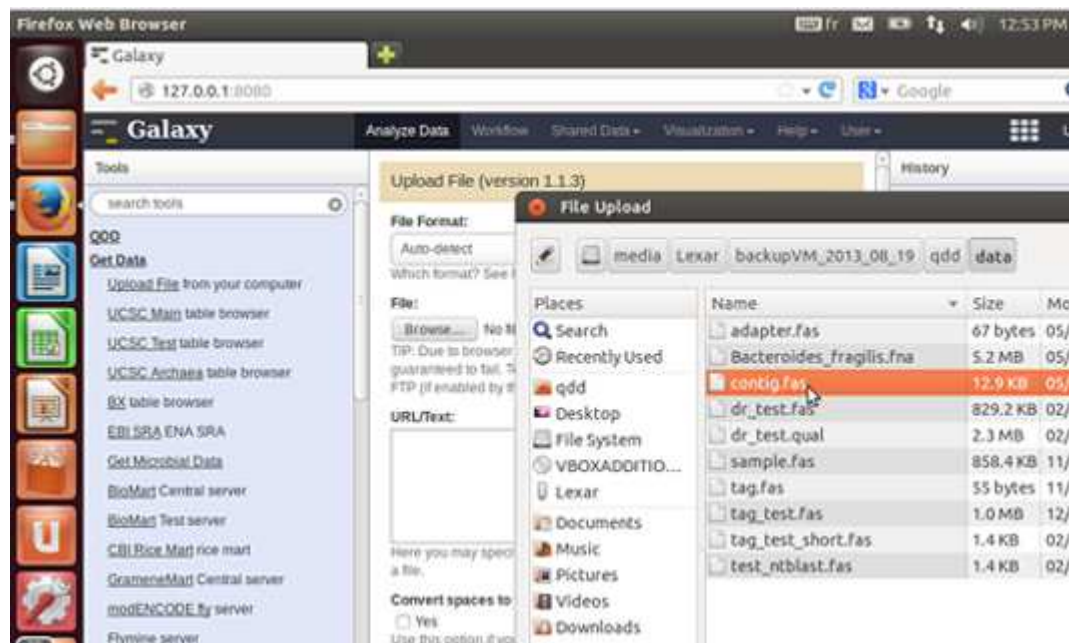


a. Devices => USB devices => your drive

From the menu Devices/USB devices select the external drive with your data on. The drive will be available to the VM but not for your host machine.



b. To get data to galaxy directly from your external drive, use the browse button to find your drive in /media/.

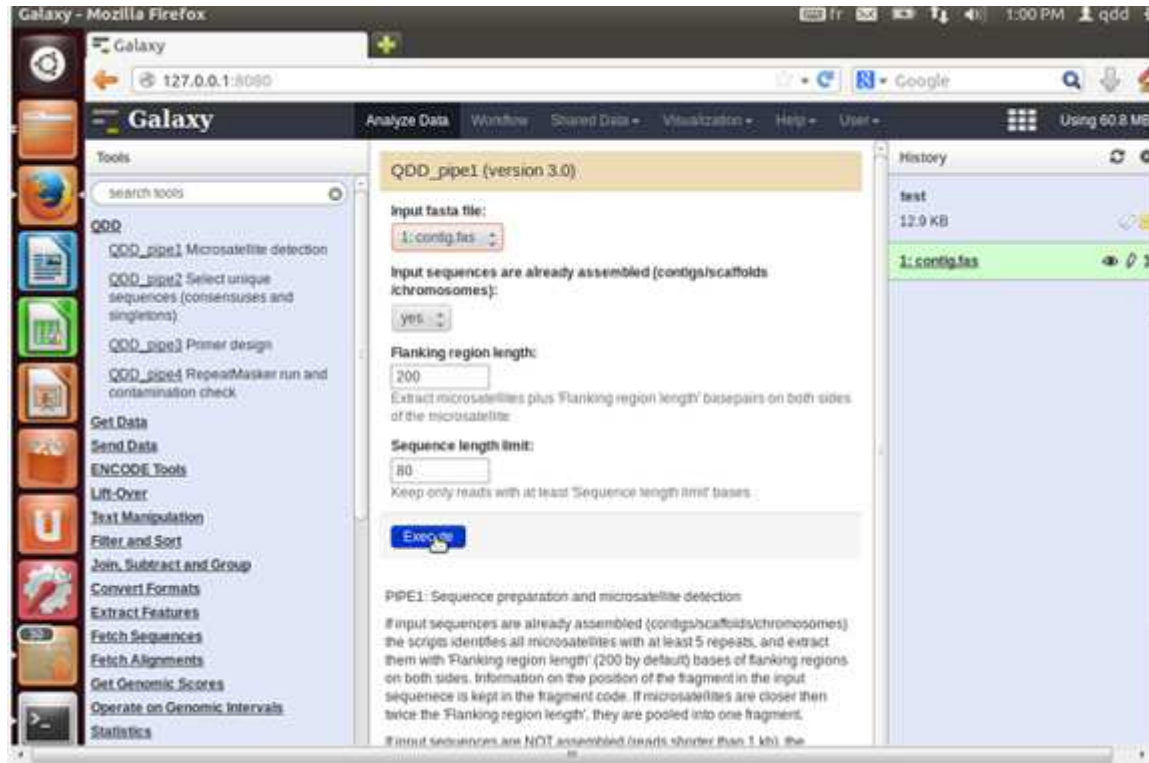


6. You are ready to run QDD. You can either run a workflow or run the four pipes one after the other.

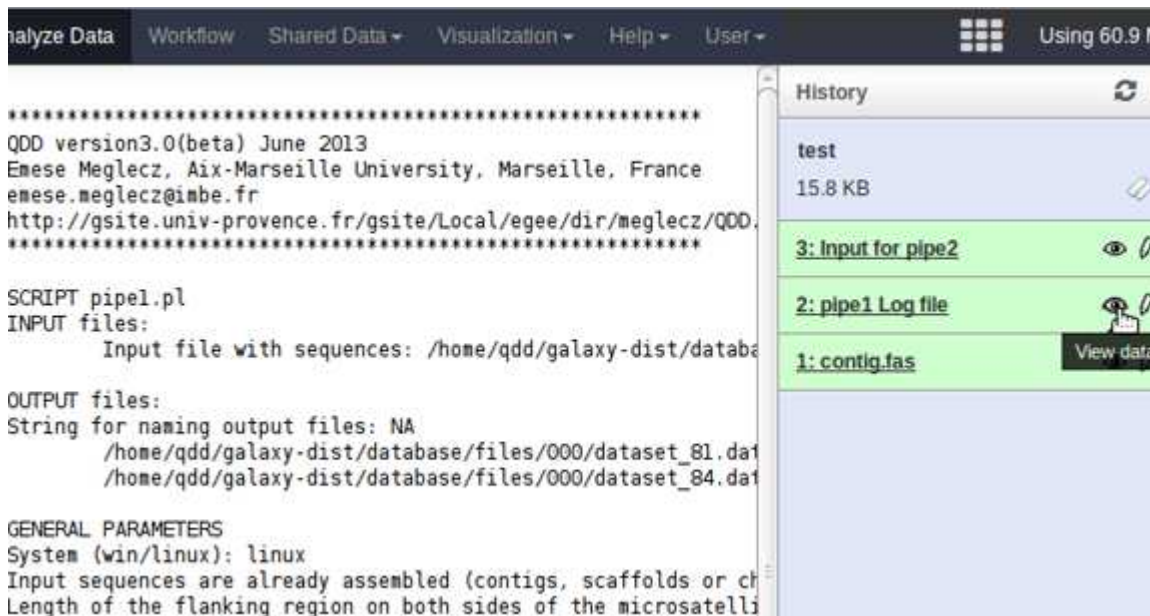
7. Running QDD pipes one by one

- Select pipe1 from the QDD menu on the left panel and set the input parameters in the middle panel and execute the program.
- The input fasta file is compulsory.

- Choose the sequence type (contigs or reads). This will alter the parameters you need to set.
- The help at the bottom of the middle panel gives a short description of pipe1.



- Once the run is finished, the output files are found in the right panel.
  - You can check them by clicking on the eye icon next to the file name. The beginning of the file appears in the middle panel.



- You can rename them by clicking on the pencil next to the file name.
- You can download them by clicking on the file name and then on the download icon that appears.
- QDD produces more files than the ones that appear by default. If you want to see all of them, you can click on the dented wheel icon (top right) and select

the 'Unhide Hidden Dataset' option.

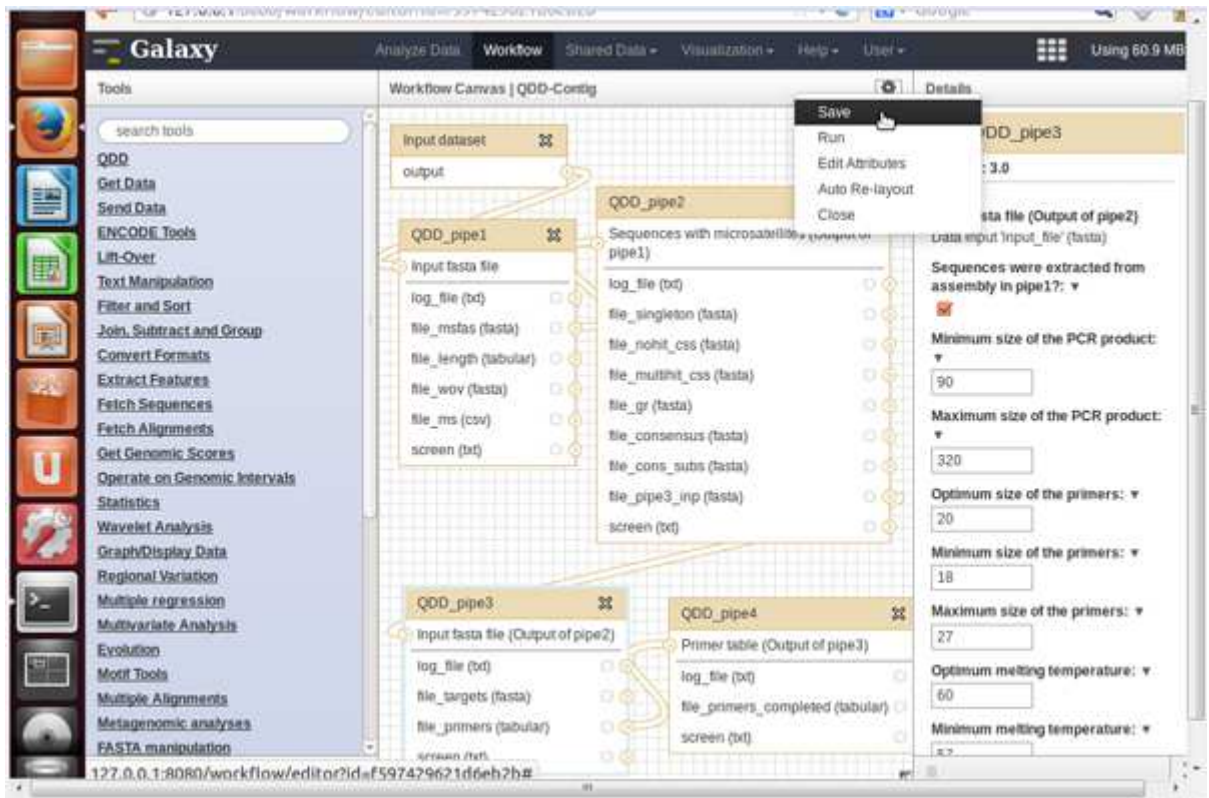
- You can run pipe2, 3 and 4 in the same way.
  - The input file for the pipe2 is the output of pipe1 **Input for pipe2**.
  - The input file for the pipe3 is the output of pipe3 **Input for pipe3**'
  - The input file for the pipe4 is the **Table with primers** produced by pipe3.
- The most important output files are
  - **Table with primers** and **Table with primers, RepeatMasker and NCBI BLAST info** which are a tab delimited tables that contains primer pairs and a lot of supplementary information to help you to choose the markers and the primers that best suite you. See for details in the [Output files](#) section. These files can be easily opened in excel once downloaded.
  - Sequences with primers.
  - Do not neglect the log files, that contain all the input parameters and summary information on the results.

#### 8. Running QDD as a workflow

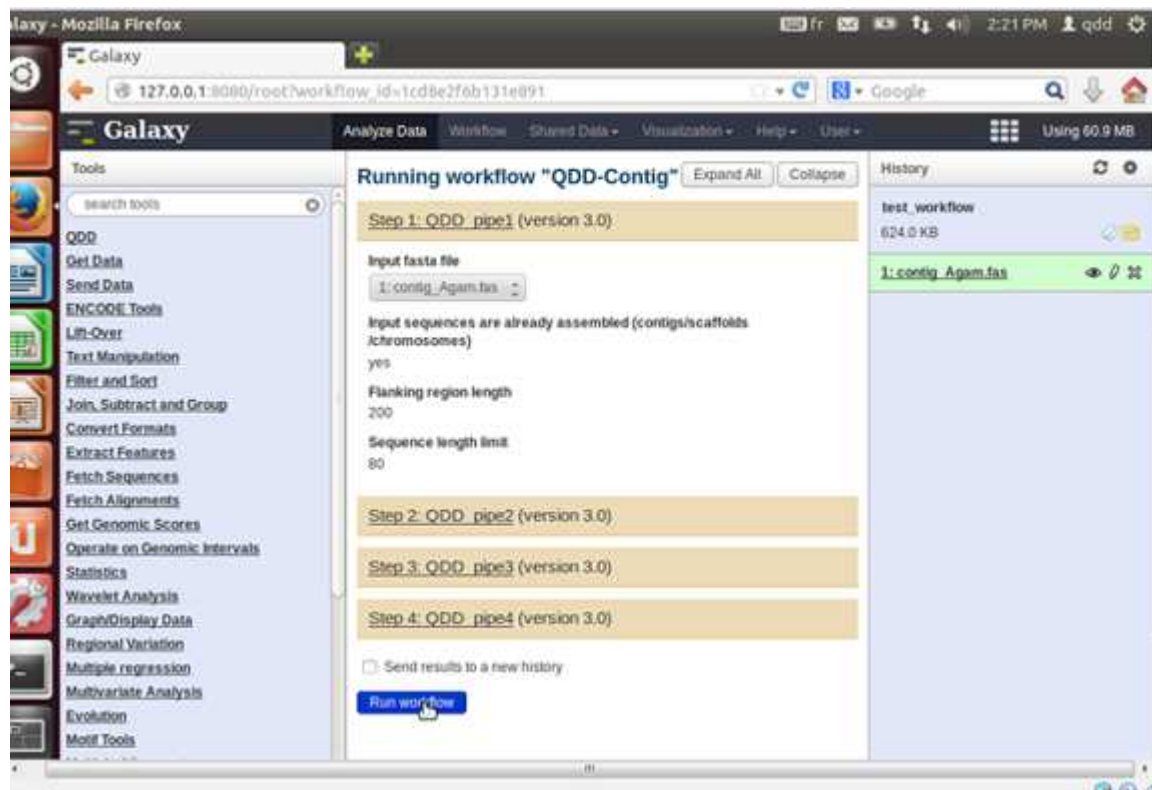
- You can access your workflows by choosing the **Workflow** in the menu on the top of the page. For using workflows, you have to be logged in (**User** in the top menu).
- You can edit or run a workflow by clicking on the triangle next to its name



- When editing the workflow, click on the block representing the step you want to edit. In the right panel you can change the input parameters. Do not forget to save your modifications (top right).



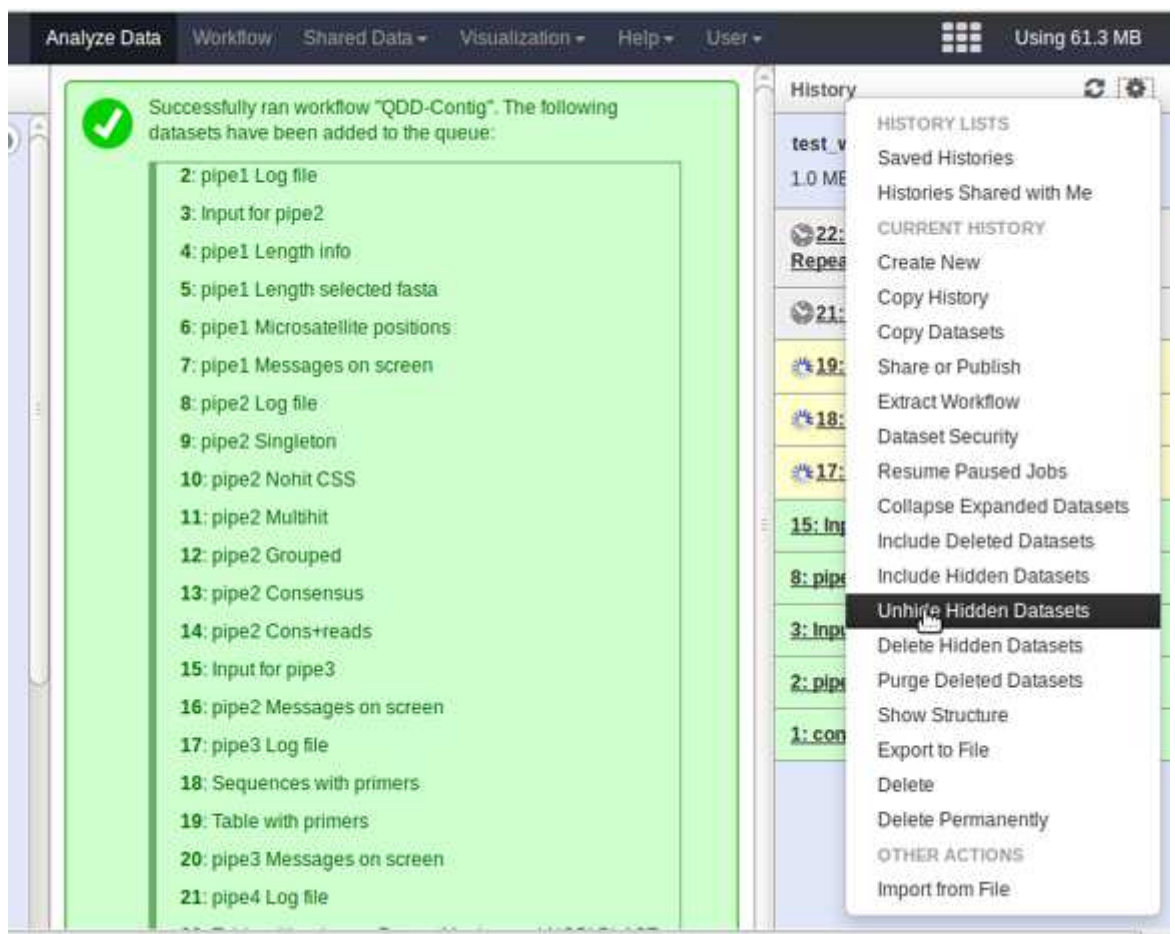
- Once edited, you can run the workflow.
  - Select a history, or make a new one with your input files in it.
  - Choose the Workflow menu (in black top menu bar), select the workflow you want to run, and select run.
  - In the middle panel you can check again the input parameters, but you cannot modify them.



- The output files appear in the right panel.
9. When it seems too long...
- Galaxy sometimes appear to be blocked, but usually it is only the right lane panel that is not refreshed. You can click to the double arrow icon on the top right to refresh this panel.



- While a script is running, you can see the output files in yellow, with a turning wheel showing that galaxy is working. If you would like more information, you can unhide the hidden files, and look at the file **pipeX messages on screen** to have more information on the steps being executed. This file can also contain error messages if something turns wrong.



- If you have plenty of sequences with primers, pipe4 can take hours or days to finish. The most important results you already have in the **Table with primes**. Pipe4 will complete this file with RepeatMasker and NCBI BLAST information. It is up to you to decide how important it is for you to get this supplementary information.

[Back to Top](#)

## Running QDD on command line

Pipe1-4 can be run separately or all in one go. In both cases, default parameters are read from `set_qdd_default.ini` file but they can be overwritten by using command line options.

See [examples](#) below.

### Running pipe1-4 separately

1. Open a terminal  
Help [windows](#) (START =>Program =>Accessories => Command Prompt), Help [linux](#)
2. Change directory in a terminal to the qdd folder (that contains the scripts; e.g. `cd d:\QDD`)
3. Make sure that the `out_folder` in `set_qdd_default.ini` is set to an existing folder. If not, modify the setting or create the folder.
4. Run `pipe1.pl`, `pipe2.pl`, `pipe3.pl` and `pipe4.pl`
  - The general syntax for running these scripts is  

```
perl pipeX.pl -parameter_name parameter_value
```
  - The `-input_file` option is compulsory, all others are optional.
  - If a parameter is not specified in the command line, the default value specified in the `set_qdd_default.ini` file is used

See [examples](#) below.

[Back to Top](#)

## QDD.pl

### Run all pipes in one go / batch submission / sorting sequences by tags

QDD.pl runs the four pipes one after the other, handles batch submission and can sort sequences in the input files according to tags. The tag sorting option is available only in command line option and not in QDD-Galaxy.

The general syntax is

```
perl QDD.pl -parameter_name parameter_value
```

1. **Batch submission:** in QDD.pl instead of one input file (`-input_file`) an input folder should be set (`-input_folder`).

This enables users to run many files in one go without giving each file name separately.

- The `input_folder` should contain all and only the input files (without the adapter or tag file) and they will be run one after the other.
- You have to use `-input_folder` even if you have only one input file.
- The `-input_file` option does not exist in QDD.pl.

2. The option `-run_all` set to 1 prompts QDD to run all 4 pipes one after the other for all files in the input folder.

If `-run_all` is 0 only the tag sorting is done (see below)

```
perl QDD.pl -input_folder data/ -run_all 1
```

3. The option `-tag` set to 1 prompts QDD to sort sequences in the input file(s) according to

tags.

In this case `-tag_file` should be set to the name of the fasta file (including path) containing all tags.

4. Apart from the `-input_file` and `-outfile_string` parameters, all other [parameters](#) described for pipe1-4 are also valid for QDD.pl

See [examples](#) below.

[Back to Top](#)

## Examples for running QDD from the command line

### Example1

You have an [assembly](#) (there might just be contigs) of an [insect](#) genome and you want to compare the sequences with successful primer design to [known transposable elements](#). Since you have done your assembly correctly, you do not need to check the contamination.

You have set the different paths in the `set_defalut_qdd.ini`, but let all the other default values:

Download input and output files of example1 [here](#).

```
perl pipe1.pl -input_file c:\qdd_data\example1.fas -contig 1
```

- Microsatellites are extracted with 200 bp flanking regions on both sides and found in the `c:\qdd_output\example1_pipe1_for_pipe2.fas`. You can change the flanking region length by setting `-flank_length`

```
perl pipe2.pl -input_file c:\qdd_output\example1_pipe1_for_pipe2.fas -make_cons 0
```

- Since you started from an assembly, it does not make sense to make consensus sequences (`-make_cons 0`). Sequences are, however, compared to each other and only the ones with no similarity to the others are kept, to avoid paralogs. The unique sequences are found in `c:\qdd_output\example1_pipe2_for_pipe3.fas`

```
perl pipe3.pl -input_file c:\qdd_output\example1_pipe2_for_pipe3.fas -contig 1
```

- After the iterative primer design, the sequences with primers are found in `c:\qdd_output\example1_pipe3_targets.fas`, and the Primer table in `c:\qdd_output\example1_pipe3_primers.tabular`.
- `-contig 1` tells QDD that the sequences have been extracted from assemblies, and it adds two columns to the primer table with the Id of the contig and the first position of the extracted fragment on the contig. When selecting markers, you should avoid closely linked markers.

```
perl pipe4.pl -input_file c:\qdd_output\example1_pipe3_primers.tabular -rm 1 -rm_lib insecta
```

- The sequences with primers are screened by RepeatMaster (`-rm 1`) against the know transposable elements of insects (`-rm_lib insecta`). You can choose almost any clades for `rm_lib`. On starting pipe4, QDD checks if the name of the group is valid, and gives you suggestions for a valid name, if it is not.

**These four steps can be done all at once by running QDD.pl**

```
perl QDD.pl -input_folder c:\data_example1 -contig 1 -make_cons 0 -rm 1 -rm_lib insecta
```

- The default of `-run_all` is 1, so you do not need to specify it. It runs all four pipes, one after the other.

- Since the default value of `-check_contamination` is 0 in `pipe4`, sequences will not be blasted against the NCBI nucleotide database.
- `-input_file` is not valid in `QDD.pl`, you should specify the input folder instead (`-input_folder c:\data_example1`), that contains the input file(s) but nothing else. If there are several files in the input folder, they are all analysed one after the other.

[Back to Top](#)

## Example2

You have **454 reads** in a fasta file. Adapters have already been removed from your sequences. You would like to **check contamination** by blasting the putative markers against genbank as a **remote BLAST**, since you have not downloaded the nt databases of the NCBI. You do **NOT** want to screen for transportable elements, since (i) you are working on windows (ii) and you have an exotic taxonomic group where there is little info on existing transposable elements anyway.

You have set the different paths in the `set_defalut_qdd.ini`, but let all the other default values:

Download input and output files of example2 [here](#).

```
perl pipe1.pl -input_file c:\qdd_data\example2.fas
```

- Sequences that contain a microsatellite and longer than 80 bp (`-length_limit 80` by default) are found in `c:\qdd_output\example2_pipe1_for_pipe2.fas`

```
perl pipe2.pl -input_file c:\qdd_output\example2_pipe1_for_pipe2.fas
```

- The unique sequences ( singletons and consensus sequences) are found in `c:\qdd_output\example2_pipe2_for_pipe3.fas`

```
perl pipe3.pl -input_file c:\qdd_output\example2_pipe2_for_pipe3.fas
```

- `c:\qdd_output\example2_pipe3_primers.tabular` contains information on the primer pairs, sequences, and target regions.

```
perl pipe4.pl -input_file c:\qdd_output\example2_pipe3_primers.tabular -check_contamination 1
```

- The sequences with primers are BLASTed against the nt database of NCBI by remote BLAST (`-local_blast` is 0 by default). This need a good internet connection, and a lot of time.
- Info on the best hits to nt is added to the primer table (`example2_pipe3_primers.tabular`) and found in `c:\qdd_output\example2_pipe4_primers.tabular`.

**These four steps can be done all at once by running QDD.pl**

```
perl QDD.pl -input_folder c:\data_example2 -check_contamination 1
```

- The default of `-run_all` is 1, so you do not need to specify it. It runs all 4 pipes, one after the other.
- `-input_file` is not valid in `QDD.pl`, you should specify the input folder instead (`-input_folder c:\data_example2`), that contains the input file(s) but nothing else. If there are several files in the input folder, they are all analyzed one after the other.

[Back to Top](#)



## Example3

You have one or more files with **454 reads** that contain **tags** at the beginning of the sequences that identify the origin of the sequence, and thus sequences need to be sorted into separate files according to tags.

You have **adapters** to be removed from your sequences (after sorting them by tag)

You would like to **check contamination** by blasting the putative markers against the nt database of ncbi, that you have downloaded and extracted on your computer and set the name and the location of this database (-blastdb) in the set\_defalut\_qdd.ini as well as **-local\_blast to 1**.

You have set the different paths in the set\_defalut\_qdd.ini, but let all the other default values (except for **-local\_blast 1**).

Download input and output files of example3 [here](#).

Tag sorting step can be done only by QDD.pl and not by pipe1.pl

```
perl QDD.pl -input_folder c:\data_example3 -tag 1 -tag_file c:\myfolder\tag.fas -adapter 1
-adapter_file c:\myfolder\adapter.fas -check_contamination 1
```

- -input\_file is not valid in QDD.pl, you should specify the input folder instead (-input\_folder c:\data\_example3), that contains the input file(s) but nothing else.
- Sequences in the fasta files are sorted according to the tags (-tag 1) that are found in c:\myfolder\tag.fas.  
Beware! The tag.fas is NOT in the input folder (c:\data\_example3)
- Then each of the resulting files are analyzed by pipe1-4, since the default of -run\_all is 1
- -adapter 1 prompts the program to clip the adapters (c:\myfolder\adapter.fas) from the sequences.  
Beware! The adapter.fas is NOT in the input folder (c:\data\_example3)
- -check\_contamination prompts the program to blast the sequences with markers (c:\qdd\_output\xxx\_targets.fas) against the nt database you have downloaded (you have set -local\_blast to 1 in the set\_qdd\_deafult.ini file).

[Back to Top](#)

## Example4

You have Illumina or Ion Torrent **low coverage data in fastq format**, thus assembling the reads does not make sense. You have trimmed off low quality regions of the reads.

You would like to **check contamination** by blasting the putative markers against genbank as a **remote BLAST**, since you have not downloaded the nt databases of the NCBI, and you would also like to compare the sequences with successful primer design to **known transposable elements** of vertebrates.

You have set the different paths in the set\_defalut\_qdd.ini, but let all the other default values:

Download input and output files of example4 [here](#).

```
perl pipe1.pl -input_file c:\qdd_data\example4.fas -fastq 1
```

- -fastq 1 will prompt QDD to convert the input fastq file to fasta
- Sequences that contain a microsatellite and longer than 80 bp (-length\_limit 80 by default) are found in c:\qdd\_output\example4\_pipe1\_for\_pipe2.fas

```
perl pipe2.pl -input_file c:\qdd_output\example4_pipe1_for_pipe2.fas
```

- The unique sequences ( singletons and consensus sequences) are found in  
c:\qdd\_output\example4\_pipe2\_for\_pipe3.fas

```
perl pipe3.pl -input_file c:\qdd_output\example4_pipe2_for_pipe3.fas
```

- c:\qdd\_output\example2\_pipe3\_primers.tabular contains information on the primer pairs, sequences, and target regions.

```
perl pipe4.pl -input_file c:\qdd_output\example4_pipe3_primers.tabular -check_contamination 1 -rm 1 -rm_lib vertebrates
```

- The sequences with primers are BLASTed against the nt database of NCBI by remote BLAST (-local\_blast is 0 by default). This need a good internet connection, and a lot of time.
- The sequences with primers are screened by RepeatMaster (-rm 1) against the know transposable elements of vertebrates (-rm\_lib vertebrates). You can choose almost any clades for rm\_lib. On starting pipe4, QDD checks if the name of the group is valid, and gives you suggestions for a valid name, if it is not.
- Information on the best hits to nt and to the Repeatmarker database is added to the primer table (example4\_pipe3\_primers.tabular) and found in  
c:\qdd\_output\example4\_pipe4\_primers.tabular.

**These four steps can be done all at once by running QDD.pl**

```
perl QDD.pl -input_folder c:\data_example4 -fastq 1 -check_contamination 1 -rm 1 -rm_lib vertebrates
```

- The default of -run\_all is 1, so you do not need to specify it. It runs all 4 pipes, one after the other.
- -input\_file is not valid in QDD.pl, you should specify the input folder instead (-input\_folder c:\data\_example4), that contains the input file(s) but nothing else. If there are several files in the input folder, they are all analyzed one after the other.

[Back to Top](#)

## List of parameters (Set in the set\_qdd\_default.ini file or on the command line)

\* These parameter values are unlikely to be changed between runs. You should set them in the *set\_qdd\_default.ini* file

Parameter name	Parameter value	Explanations	default	pipe1	pipe2	pipe3	pipe4
-input_file	Fasta file with full path	Fasta file sequences to be analyzed for 1-3; Primer table for pipe4	NA	yes	yes	yes	yes
General parameters							
-galaxy	0 or 1	1 for running QDD form Galaxy, 0 for running it from a terminal	0	yes	yes	yes	yes
-syst	win or linux	Operating system	* linux	yes	yes	yes	yes
-blast_path	Full path to BLAST+		*	yes	yes	no	yes

	executables						
-clust_path	Full path to clustalw executables		*	no	yes	no	no
-primer3_path	Full path to Primer3 executables		*	no	no	yes	no
-primer3_version	1 or 2	Version of Primer3	*2	no	no	yes	no
-qdd_folder	Full path to QDD scripts		*	yes	yes	yes	yes
-out_folder	Folder for output files	Folder (with full path) for output files. Must be created before running qdd	*	yes	yes	yes	yes
-del_files	0 or 1	1 for deleting temporary files after the run1	* 1	yes	yes	yes	yes
-outfile_string	alpha-numerical	String to specify the beginning of the names of the output files. If not specified, the input file name is used for naming output files		yes	yes	yes	yes
-debug	0 or 1	1 for printing out more details in log file for debugging	* 0	yes	yes	yes	yes
-num_threads	integer	number of threads (CPU) for BLAST and RepeatMasker	* 1	yes	yes	no	yes
<b>Input sequence/file type</b>							
-fastq	0 or 1	1 if input file is in fastq format, 0 if it is in fasta format	0	yes	no	no	no
-contig	0 or 1	1 if sequences has been assembled (contigs, scaffolds, chromosomes); 0 if they are short (100-1000) sequencing reads; If -contig=1 -adapter must be 0	0	yes	no	yes	no
-flank_length	integer	Length of the flanking region on both sides if the microsatellite, when extracting microsatellites from contigs (-contig=1)	* 200	yes	no	no	no
<b>Adapter clipping</b>							
-adapter	0 or 1	1 for running adaptor/vector clipping step If -adapter = 1 => -adapter_file must be set.	0	yes	no	no	no
-adapter_file	fasta file (including full path)	File that contains the adapters to be removed from the sequences. For adapter clipping -adapter must be set to 1, it is not enough to set only the -adapter_file	NA	yes	no	no	no
<b>Sequence length selection</b>							
-length_limit	integer	Sequences shorter then length_limit (after adapter clipping) are	80	yes	no	no	no

		eliminated					
<b>Consensus</b>							
-make_cons	0 or 1	Make consensus sequences	1	no	yes	no	no
-ident_limit	[80-100]	Minimum % of pairwise identity between sequences of a contig (only if -make_cons=1)	95	no	yes	no	no
-prop_maj	[0.5-1]	Proportion of sequences that must have the same base at a site to accept it as a consensus (only if -make_cons=1)	0.66	no	yes	no	no
<b>RepeatMasker (checking similarity to transposable elements)</b>							
-rm	0 or 1	1 for running RepeatMasker on the sequences with primers (not available for windows)	0	no	no	no	yes
-rm_path	Full path to RepeatMasker executables		*	no	no	no	yes
-rm_lib	Taxonomic group	A taxonomic group for which TE library is selected RepeatMasker e.g. metazoa, vertebrata, insecta...	eukary ota	no	no	no	yes
<b>Compare sequences to NCBI</b>							
-check_contamination	0 or 1	1 for checking contamination by blasting sequences against the nt database (blast can be remote or local)	0	no	no	no	yes
-local_blast	0 or 1	1 for running local blast for contamination check => blastdb needs to be defined; 0 for running remote blast for contamination check	* 0	no	no	no	yes
-blastdb	name (including full path) of a local ncbi database	Only needed if local BLAST is used for contamination check (-local_blast=1)	*	no	no	no	yes
<b>Primer design</b>							
-pcr_min	[40,10000]	Minimum PCR Product size	90	no	no	yes	no
-pcr_max	[40,10000]	Maximum PCR Product size	300	no	no	yes	no
-pcr_step	[20,10000]	PCR product size interval for iterative primer design	50	no	no	yes	no
-PRIMER_GC_C LAMP	[0,20]	Require the specified number of consecutive Gs and Cs at the 3' end of both the left and right primer	0	no	no	yes	no
-PRIMER_OPT_SIZE	[1,50]	Optimum length of a primer	20	no	no	yes	no

- PRIMER_MIN_SIZE	[1,50]	Minimum length of a primer	18	no	no	yes	no
- PRIMER_MAX_SIZE	[1,50]	Maximum length of a primer	27	no	no	yes	no
- PRIMER_OPT_TM	[1,100]	Optimum melting temperature	60	no	no	yes	no
- PRIMER_MIN_TM	[1,100]	Minimum melting temperature	57	no	no	yes	no
- PRIMER_MAX_TM	[1,100]	Maximum melting temperature	63	no	no	yes	no
- PRIMER_MAX_DIFF_TM	[1,100]	Maximum acceptable difference between the melting temperatures of primers	10	no	no	yes	no
- PRIMER_MIN_GC	[1,100]	Minimum percentage of Gs and Cs in any primer	20	no	no	yes	no
- PRIMER_OPT_GC_PERCENT	[1,100]	Optimum percentage of Gs and Cs in any primer	50	no	no	yes	no
- PRIMER_MAX_GC	[1,100]	Maximum percentage of Gs and Cs in any primer	80	no	no	yes	no
- PRIMER_SELF_ANY	[1,100]	The maximum allowable local alignment score for self- or pairwise-complementarity	8	no	no	yes	no
- PRIMER_SELF_END	[1,100]	The maximum allowable 3'-anchored global alignment score for self- or pairwise-complementarity	3	no	no	yes	no
- PRIMER_MAX_POLY_X	[1,10]	The maximum allowable length of a mononucleotide repeat in primer	3	no	no	yes	no
- PRIMER_NUM_RETURN	[1,10]	The maximum number of primer pairs to return for each Primer3 run (each PCR product length interval each stringency step on target region)	3	no	no	yes	no
QDD.pl							
-input_folder	Folder name including path	Folder that contains all input files for batch submission; must not contain other files	NA	QDD.pl	QDD.pl	QDD.pl	QDD.pl
-tag	0 or 1	1 for sorting input sequences according to tags	0	QDD.pl	QDD.pl	QDD.pl	QDD.pl
-tag_file	Fasta file including full path	fasta file with tag sequences	NA	QDD.pl	QDD.pl	QDD.pl	QDD.pl
-run_all	0 or 1	1 for running pipe1, pipe2, pipe3 and pipe4 in one go	1	QDD.pl	QDD.pl	QDD.pl	QDD.pl



# Output files

## Naming output files in command line version

All output files are found in the output folder (-out\_folder). If the -outfile\_string is not specified, the name of the input file is used for naming the output files followed by a complementary string to refer to the content of the file. (e.g. input : sample.fas output: sample\_pipe1\_log.txt). If -outfile\_string is set (e.g. -outfile\_string test) a complementary string to referring to the content of the output file is attached to the -outfile\_string (e.g. test\_pipe1\_log.txt).

If the output file name already exists, files will be numbered e.g. sample\_pipe1\_log\_v1.txt

[Back to Top](#)

## Output files of PIPE1

File names in Galaxy are in parentheses

1. **xxx\_pipe1\_log.txt (pipe1 Log file)** => Parameters of the run and summary
2. **xxx\_pipe1\_for\_pipe2.fas (Input for pipe2)** => Fasta file with microsatellite containing sequences, longer than a preset limit (-length\_limit) after adapter clipping; It is the input file for pipe2.pl
3. **xxx\_pipe1\_wov.fas (pipe1 Length selected fasta)** => Fasta file with sequences longer than -length\_limit after the adapter clipping
4. **xxx\_pipe1\_ms.csv (pipe1 Microsatellite positions)** => File with information on microsatellite motifs and positions in each sequences. Semicolons are used for separating columns
  - 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
5. **xxx\_pipe1\_length\_info.tabular (pipe1 Length info)** => Information on sequence length and adapter clipping; tab separated columns
  - Column1: Sequence code
  - Column2: Original length of the sequence
  - Column3: Number of bases cut from the beginning of the sequence
  - Column4: Number of bases cut from the end of the sequence
  - Column5: Length of the sequence after cutting adapter/vector

[Back to Top](#)

## Output files of PIPE2

File names in Galaxy are in parentheses

1. **xxx\_pipe2\_log.txt (pipe2 Log file)** => Parameters of the run and summary
2. **xxx\_pipe2\_for\_pipe3.fas (Input for pipe3)** => Fasta file with all unique sequences (singletons and consensus); Input file for pipe3.pl
3. **xxx\_pipe2\_singleton.fas (pipe2 Singleton)** => Fasta file with singletons (the only BLAST hit is autohit)
4. **xxx\_pipe2\_nohit\_css.fas (pipe2 Nohit CSS)** => Fasta file with low complexity sequences (no BLAST hit to itself)
5. **xxx\_pipe2\_multihit\_css.fas (pipe2 Multihit)** => Fasta file with putatif minisatellites (more than one hit (local alignment) between a pair of sequences)
6. **xxx\_pipe2\_grouped.fas (pipe2 Grouped)** => Fasta file with sequences (including consensus) that had BLAST hit to other sequences, with bellow limit identity of the overlapping region. It can be either a partial similarity (only a region of the two sequences can be aligned), or a the two sequences are aligned in their entire length, but the percentage of similarity is bellow limit. Regions covered by BLAST hits are masked by lower case letters
7. **xxx\_pipe2\_consensus.fas (pipe2 Consensus)** => Fasta file with all unique (no hit to grouped sequences) consensus 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.
8. **xxx\_pipe2\_cons\_subs.fas (pipe2 Cons+reads)** => Fasta file with consensus + aligned reads to make a consensus

[Back to Top](#)

## Output files of PIPE3

File names in Galaxy are in parentheses

1. **xxx\_pipe3\_log.txt (pipe3 Log file)** => Parameters of the run and summary
2. **xxx\_pipe3\_targets.fas (Sequence with primers)** => Sequences with successful primer design
3. **xxx\_pipe3\_primers.tabular (Table with primers)** => Information on primers, target regions, primers...
  - SEQUENCE\_CODE: Original codes for singletons and cons\_grX\_Y codes for consensus sequences
  - NUMBER\_OF\_READS: The number of reads underlying the sequence. 1 for singletons, >1 for consensus.
  - 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 targeted, the target region covers the microsatellite (compound or pure). Otherwise the target 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 number. If microsatellite is compound, it is the number of repetition in the longest uninterrupted stretch. If there are more than one



microsatellites in the target region, target MS info refers to the longest (in repeat numbers) of the target microsatellites.

- **NUMBER\_OF\_MS**: The number of microsatellites in the target region. 1 for one pure microsatellite, 1.5 for one compound microsatellite, >1.5 the number of microsatellites (regardless whether pure and compound)
- **MOT\_TRANS**: Repeat motif type, where circular permutations and their reverse complementary sequences are pooled (e.g. AC refers to AC, CA, TG, GT). If there are more than one microsatellite in the target region, it refers to the longest (in repeat numbers) of the target microsatellites.
- **TARGET\_REGION\_SEQ**: Sequence of the target region as found in the read/consensus
- **POLYMORPH**: If polymorphism is detected, then the repeat motif and its position is indicated. NA for singletons, NO if the MS has the same length in all reads of a consensus.
- **ONE\_PRIMER\_FOR\_EACH\_SEQ**: Only one primer is selected for each sequence. Selecting lines with 1 in this column gives the total number of sequences with primers. The selection between the “best primer pairs” of each target region is based on the number of microsatellites in the target region (**NUMBER\_OF\_MS**; the lowest the better) and the length of the microsatellite (**TARGET\_MS\_LENGTH\_IN\_REPEAT\_NUMBER**; the highest the better). This ordering is based on lab tests of PCR success rate and polymorphism of different primers (Meglecz et al. Submitted).
- **ONE\_PRIMER\_FOR\_EACH\_TARGET\_REGION**: Only one primer is selected for each target region. Selecting lines with 1 in this column gives the total number of target regions with primers. There can be more than one target region per sequence, so some of the markers are strongly linked. The selection is based on the alignment score between the primers and the amplicon (**PCR\_PRIMER\_ALIGNSCORE**; the lowest the better), on the distance between primer and the target region (**MIN\_PRIMER\_TARGET\_DIST**; the highest the better) and the size of the PCR product (**PCR\_PRODUCT\_SIZE**; the lowest the better). This ordering is based on lab tests of PCR success rate of different primers (Meglecz et al. Submitted)
- **PCR\_PRIMER\_ALIGNSCORE**: The maximum alignment score between the primers and the sequence (excluding primers; from version 3.1.2) The maximum alignment score between the primers and the amplicon excluding primers (versions 3.1 and 3.1.1)
- **MIN\_PRIMER\_TARGET\_DIST**: The smallest distance between the 3' end of the two primers and the target region. If primer(s) match more than once the sequence, the distance is calculated for the annealing site closer to the target region.
- **PCR\_PRODUCT\_SIZE**: PCR product size in bp including primers. If primer(s) match more than once the sequence, the size of the longest PCR product size is given here.
- **PCR\_PRODUCT\_SEQ**: Sequence of amplicon including primers. If primer(s) match more than once the sequence, the longest PCR product is given here.
- **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 right 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
- PRIMER3\_PENALTY: Primer pair penalty (see documentation of Primer3)
- DESIGN: guides for target region complexity

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	+	+	+	+	+

+ allowed, but not necessarily present; - not allowed

- SEQUENCE\_LENGTH: Length of the read or consensus
- SEQUENCE: the whole sequence with homopolymers micro- and nanosatellites printed in lower case
- CONTIG\_CODE (If contig = 1): id of the contig
- FIRST\_POS\_ON\_CONTIG (If contig = 1): First position of the extracted fragment on its contig. These last two columns help to avoid choosing markers too close to each other on the same contig

[Back to Top](#)

## Output files of PIPE4

File names in Galaxy are in parentheses

1. **xxx\_pipe4\_log.txt (pipe4 Log file)** => Parameters of the run and summary
2. **xxx\_pipe4\_primers.tabular (Table with primers, RepeatMasker and NCBI BLAST hit info)** => Same information as in xxx\_pipe3\_primers.tabular, completed by
  - If check\_contamination = 1 => Information on the best hit against the nt database of NCBI (accession, description, e-value, score); Classification taxonomic of the species of the best hit (Kingdom, Phylum, Class, Family, Genus, Species)
  - If rm = 1 => Information on the best hit to the interspersed elements library.



# Trouble shooting

## QDD VM does not boot

You followed the installation process described [here](#) but at point 6 the VM does not boot.

Try to disable the USB2 controller

1. Open VirtualBox Manager
2. Select VM
3. Select [Settings](#), then [USB](#)
4. Uncheck [Enable USB 2.0 \(EHCI Controller\)](#)
5. Start VM

## Primer3 error message: 'Unable to open file primer3\_config.stack.ds'

QDD is running but you have error messages from primer3 and no primer is designed

1. Make sure you have set correctly the path to primer3\_core(.exe) in the set\_qdd\_default.ini file
2. If it does not help, move the primer3\_config folder to the directory where primer3\_core is found

## PCR product size - primer position

**PCR product size does not match with primer positions in the sequence**

The primer positions are given as it was in the output of primer3. However, in some cases, one or even both primers can match more than once the underlying sequence. In these cases, the sequence of the PCR product (PCR\_PRODUCT\_SEQ) and the length of the PCR product (PCR\_PRODUCT\_SIZE) are given for the longest PCR product. On the other hand, the primer – target region distance (MIN\_PRIMER\_TARGET\_DIST) is given for the shortest PCR product. This decision was based on our lab results where we found that longer PCR products and primer pairs close to the target regions are less likely to give a clear genotyping profile ([Meglecz et al. 2014](#)).

If primers match more than once the sequence, the PCR\_PRIMER\_ALIGNSCORE is high, and these primer pairs are very unlikely to be flagged as the "best pair" for the sequence (ONE\_PRIMER\_FOR\_EACH\_SEQ).