XML questions and data format

XML questions

Here is the proposed directory structure:

Amplicon/Fastq

Amplicon/XML

Amplicon/Colonies

Check if it's only one id starting with ERR105 under each DESCRIPTION tag

```
cd ~/Amplicon/XML
cat *.xml | grep -c "<DESCRIPTION>"
cat *.xml | grep -c "<ID>ERR105"
cat *.xml | grep -e "<DESCRIPTION>\|<ID>ERR105"
```

We can still not be sure if we don't look at the output, but that can be tiring if there are many samples. This should be safe:

```
cat *.xml | grep -e "<DESCRIPTION>\|<ID>ERR105" | paste - - | grep
    -ce "DESCRIPTION.*ERR105"
```

This should produce the same nummber as `cat *.xml | grep -c "<ID>ERR105"`. Now it's time to produce the sequence id files for the colonies. An example for one colony:

```
cat CT.A.xml | grep -e "<DESCRIPTION>\|<ID>ERR105" | grep -A 1
    termite | grep "<ID>" | cut -d \> -f 2 | cut -d \< -f 1
```

If you want to loop:

```
ls *.xml | while read file; do base=`echo $file | sed s/\.xml//`;
    cat $file | grep -e "<DESCRIPTION>\|<ID>ERR105" | grep -A 1
    termite | grep "<ID>" | cut -d \> -f 2 | cut -d \< -f 1 >
    $base.id; done
```

Then we uncompress the `fastq` files.

```
cd ~/Amplicon/Fastq
gunzip *.gz
du -sh Amplicon/Fastq # Just to see disk space ...
```

Produce a file of files, a so called fof:

```
ls *.fastq > fof.txt
```

Produce colony fastq files, one colony at the time:

```
# Forward:
grep -f ../XML/CT.A.id fof.txt | grep _1 | while read line; do cat
    $line >> ../Colonies/CT.A_1.fastq; done
# Reverse
grep -f ../XML/CT.A.id fof.txt | grep _2 | while read line; do cat
    $line >> ../Colonies/CT.A_2.fastq; done
```

If you want to loop we can append fastq files, but before we do that we have to delete any existing produced concatenated files:

```
rm ../Colonies/*
```

Then we loop:

```
ls ../XML/*.id | while read file; do base=`echo $file | cut -d \/
    -f 3 | cut -d . -f 1-2`; grep -f $file fof.txt | grep _1 |
    while read line; do cat $line >> ../Colonies/${base}_1.fastq;
    done; done

ls ../XML/*.id | while read file; do base=`echo $file | cut -d \/
    -f 3 | cut -d . -f 1-2`; grep -f $file fof.txt | grep _2 |
    while read line; do cat $line >> ../Colonies/${base}_2.fastq;
    done; done
```

Here comes a clearer syntax that we have still not used. We substitute backticks with $() syntax. This syntax allows for nesting of commands (which we are not using here):

```
rm ../Colonies/*
ls ../XML/*.id | while read file; do base=$(echo $file | cut -d \/
    -f 3 | cut -d . -f 1-2); grep -f $file fof.txt | grep _1 |
    while read line; do cat $line >> ../Colonies/${base}_1.fastq;
    done; done

ls ../XML/*.id | while read file; do base=$(echo $file | cut -d \/
    -f 3 | cut -d . -f 1-2); grep -f $file fof.txt | grep _2 |
    while read line; do cat $line >> ../Colonies/${base}_2.fastq;
    done; done
```