Input files:
- `japonica_IRGSP_7_chr.fasta`: the genome to be masked
- `PReDa_121015.fasta`: the nucleotidic repeat library
- `TE_protein_db_121015.fasta`: the protein database

Scripts/softwares:
- Blaster and Matcher (provided - inside /REPET_linux-x64_1.4/)
- RepeatAnnotation pipeline (provided)
- RepeatMasker, WUBLAST, Blastall, and Tandem Repeat Finder (not provided - installation required)

TE Protein Blastx
Blaster (http://urgi.versailles.inra.fr/Tools/Blaster) is a wrapper of blastall, breaks long molecules in chunks and aligns them to the chosen database. It runs on a single processor. The genome can be split in individual chromosomes and run in separate Blaster (ant then Matcher) processes. Of the different Blaster outputs files, the `.align` file is used by Matcher to reduce the redundancy of the results. Matcher outputs a `.tab` file, converted in a `.gff` file with an awk command.

Blaster and Matcher documentation:
/REPET_linux-x64_1.4/doc/BLASTERsuite_doc.txt
/REPET_linux-x64_1.4/doc/Blaster_documentation.pdf

Blaster command example (MUST be run from inside /REPET_linux-x64_1.4):
```
./bin/blaster -q ../path_to/genome.fasta -s ../path_to/TE_protein_db.fasta -n blastx -b ../path_to/out_blaster
```

Matcher command example (MUST be run from inside /REPET_linux-x64_1.4):
```
./bin/matcher2.25 -q ../path_to/genome.fasta -s ../path_to/TE_protein_db.fasta -m ../path_to/file.align -b ../path_to/out_matcher_bx
```

From Matcher .tab output file to gff:
```
awk 'NR > 1' file.tab | awk 'col="repeat", x="."{if ($9 > $8) s = "+"; else s = "-"; printf "%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\n",$1,col,substr($7,1,3), $2,$3,$14,s,x,$7}' |sort -k 1,1 -k 4,4 >genome_bx.gff
```

Repeat masking with nucleotidic database
RepeatMasker (http://www.repeatmasker.org/ for code and documentation) screens DNA sequences for interspersed repeats and low complexity DNA sequences. We will use a custom repeat library with its own classification system specifically designed for the downstream analysis. Set WUBLAST as the default engine. If not available, use RMBlast (http://www.repeatmasker.org/RMBlast.html), that might allow for multiple threading.

RMasker command:
```
RepeatMasker -qq -norna -pa 3 -no_is -cutoff 250 -gff -lib PReDa_121015.fasta genome.fasta
```

It will give several output files, the `.out.gff` file will be used in the RepeatAnnotation pipeline.

Length file
Is a text file with two tab-delimited columns: the first contains the header of each fasta in the `genome.fasta` file, the second the length of it. The following command uses the EMBOSS application Infoseq. It can be replaced by any hand-made script. The number of lines must be equal to the number of fastas in the genome.

```
infoseq -only -name -length genome.fasta | awk 'NR > 1' >genome.length
```

If using this application, replace the filed delimiter “space” with “tab”.

RepeatAnnotation pipeline
Merges evidence from RMasker and Blastx in a unique `.cons.gff` file and a table (`.txt`) file (see slides for examples). In addition to this, a masked version of the input genome is also required. Transposable element sequences (R__ and D__ codes in column 3 of the `.cons.gff`) will be hard masked (AGCCTAGCT →NNNNNNNNN), non-TE repeats (B__, S__, and XXX codes) will be soft masked (AGCCTAGCT → agcctagct). Some tools might be already available on the internet. If the process requires lots of time, parallelization of this step (e.g. one processor/chromosome) could be advantageous.

Pipeline command:

```
/path_to/Repeat_Annotation.pl genome.fasta.out.gff genome.length genome_bx.gff
```

The requested outputs are the `.cons.gff` file, the `.txt` file, and the `.masked.fasta` genome (not currently produced - see slides).

For questions and comments, contact:
Dario – Biological and bioinfo aspects dcopetti@cals.arizona.edu
Kapeel – Bioinfo aspects kapeel@cals.arizona.edu