VitisOmics

Timothée Flutre

October 9, 2015

Contents

1 Overview

This document describes the "VitisOmics" project. This project aims at handling "omics" data in the genus Vitis (e.g. grapevine) in an open and reproducible way. Nevertheless, it requires some basic knowledge and skills about bioinformatics on GNU/Linux computers.

Such "omics" data are already available from various places, such as URGI, NCBI, EBI, etc; and several committes from the IGGP (International Grape Genome Program) strive at improving interoperability. But my attempt, via the usage of git and GitHub, could prove for the community to be a useful addition to these efforts.

The project directory is organized as advised by Noble (PLoS Computational Biology 2009). On any Unix-like system, it is easily done with the following commands:

```
mkdir -p VitisOmics; cd VitisOmics/
touch AUTHORS COPYING README; mkdir -p doc data src results
```

On any Unix-like system, the project directory can also be easily compressed and transferred (ignoring large data files):

```
cd ..; tar -czvf VitisOmics.tar.gz \
--exclude=VitisOmics/data --exclude=VitisOmics/results \
--exclude="*~" --exclude=".*" VitisOmics
```

In order to concretely promote collaborative editing in a distributed manner, the content of the "VitisOmics" repository should be based on "plain text" files. As a consequence, this document is written in the org format, and can thus be automatically exported into the pdf format, best by emacs, but also by pandoc.

1.1 Contributors

The person roles comply with R's guidelines (The R Journal Vol. 4/1, June 2012).

- Timothée Flutre (cre,aut)
- Gautier Sarah (ctb)

2 Data

```
mkdir -p data; cd data/
```

TODO: retrieve genome data from other cultivars than PN40024, e.g. Sultanina

2.1 URGI

- `https://urgi.versailles.inra.fr/Species/Vitis/Data-Sequences/Genome-sequences`

```
mkdir -p urgi; cd urgi/
../../src/download_urgi.bash
```

When needed, the script decompresses zip files and compress them again but with gzip instead.

2.2 NCBI

- `http://www.ncbi.nlm.nih.gov/genome/401`
- `ftp://ftp.ncbi.nlm.nih.gov/genomes/Vitis_vinifera/`

```
mkdir -p ncbi; cd ncbi/
../../src/download_ncbi.bash
```

Note the important file `scaffold_names` which provides the correspondence between original scaffold names (i.e. from the sequencing center) and various NCBI identifiers (RefSeq, GenBank, etc).

In the archives, build 1.1 corresponds to the 8x genome sequence of PN40024, with the assembly in `Assembled_chromosomes/` whereas the contigs are in each `CHR_x/` directory.

2.3 EBI

12X.0 as well as soft-masking by RepeatMasker

```
mkdir -p ebi; cd ebi/
../../src/download_ebi.bash
```

3 Results

```
mkdir -p results; cd results/
```

TODO: compress fasta files with `bgzip` instead of `gzip`

On a computer cluster, indexed files could be copied for usage by everyone, e.g. in `/Genomics/Vitis` if on the CIRAD cluster of the SouthGreen platform.

We need to keep the information about the data source (e.g. URGI or NCBI) because there can be differences in terms of N spacers and additional scaffolds (from Aegilops?) at the NCBI.

3.1 URGI

```
mkdir -p urgi; cd urgi/
```

3.1.1 Reformat sequence headers for `VITVI_PN40024_12x_v0_chroms_URGI`

Launch script:

```
ln -s ../../data/urgi/VV_chr12x.fsa.gz .
echo "../../src/reformat_VV_chr12x.bash" \
  | qsub -cwd -j y -V -N reformat_VV_chr12x -q normal.q
```

Check:

```
zcat VV_chr12x.fsa.gz | wc -l # 8240706
zcat VV_chr12x.fsa.gz | grep -c ">" # 33
zcat VITVI_PN40024_12x_v0_chroms_URGI.fa.gz | wc -l # 8240706
zcat VITVI_PN40024_12x_v0_chroms_URGI.fa.gz | grep -c ">" # 33
diff <(zcat VV_chr12x.fsa.gz) <(zcat VITVI_PN40024_12x_v0_chroms_URGI.fa.gz)
```

Only the headers differ, not the sequences, so everything is fine.

Basic stats:

```
zcat VITVI_PN40024_12x_v0_chroms_URGI.fa.gz | md5sum #
   eff315994fafe35333462b9595e10ce5
```

3.1.2 Reformat sequence headers for `VITVI_PN40024_12x_v0_scaffolds_EMBL_r102`

Launch script:

```
ln -s ../../data/urgi/VV_12X_embl_102_Scaffolds.fsa.gz .
echo "../../src/reformat_VV_12X_embl_102_Scaffolds.bash" \
  | qsub -cwd -j y -V -N reformat_VV_12X_embl_102_Scaffolds -q normal.q
```

Check:

```
zcat VV_12X_embl_102_Scaffolds.fsa.gz | wc -l # 8091565
zcat VV_12X_embl_102_Scaffolds.fsa.gz | grep -c ">" # 2059
zcat VITVI_PN40024_12x_v0_scaffolds_EMBL_r102.fa.gz | wc -l # 8091565
zcat VITVI_PN40024_12x_v0_scaffolds_EMBL_r102.fa.gz | grep -c ">" # 2059
diff <(zcat VV_12X_embl_102_Scaffolds.fsa.gz) <(zcat
   VITVI_PN40024_12x_v0_scaffolds_EMBL_r102.fa.gz)
```

Only the headers differ, not the sequences, so everything is fine.

Basic stats:

```
zcat VITVI_PN40024_12x_v0_scaffolds_EMBL_r102.fa.gz | md5sum # 4
   fa2432d7a66c019c7cb41ee4d0cb7bc
```

3.1.3 Reformat sequence headers for `VITVI_PN40024_12x_v0_contigs_EMBL_r102`

TODO

3.1.4 Reformat sequence headers for `VITVI_PN40024_12x_v2_chroms_URGI`

Launch script:

```
ln -s ../../data/urgi/12Xv2_grapevine_genome_assembly.fa.gz .
echo "../../src/reformat_12Xv2_grapevine_genome_assembly.bash" \
  | qsub -cwd -j y -V -N reformat_12Xv2_grapevine_genome_assembly -q normal.q
```

Check:

```
zcat 12Xv2_grapevine_genome_assembly.fa.gz | wc -l # 8103449
zcat 12Xv2_grapevine_genome_assembly.fa.gz | grep -c ">" # 20
zcat VITVI_PN40024_12x_v2_chroms_URGI.fa.gz | wc -l # 8103449
zcat VITVI_PN40024_12x_v2_chroms_URGI.fa.gz | grep -c ">" # 20
diff <(zcat 12Xv2_grapevine_genome_assembly.fa.gz) <(zcat
   VITVI_PN40024_12x_v2_chroms_URGI.fa.gz)
```

Only the headers differ, not the sequences, so everything is fine.

Basic stats:

```
zcat VITVI_PN40024_12x_v2_chroms_URGI.fa.gz | md5sum # 4
    e487c28eaf19ef59b0b6128b73935af
```

Length of each sequence:

```
zcat VITVI_PN40024_12x_v2_chroms_URGI.fa.gz \
  | awk 'BEGIN{RS=">"} {split($0,a,"\n");
if(length(a)==0)next;
sum=0; for(i=2;i<=length(a);++i){sum+=length(a[i])};
print a[1]": "sum; sumTot+=sum} END{print sumTot}'
```

header	length (bp)
chr1 Vitis vinifera|PN40024|assembly12x.2	24233538
chr2 Vitis vinifera|PN40024|assembly12x.2	18891843
chr3 Vitis vinifera|PN40024|assembly12x.2	20695524
chr4 Vitis vinifera|PN40024|assembly12x.2	24711646
chr5 Vitis vinifera|PN40024|assembly12x.2	25650743
chr6 Vitis vinifera|PN40024|assembly12x.2	22645733
chr7 Vitis vinifera|PN40024|assembly12x.2	27355740
chr8 Vitis vinifera|PN40024|assembly12x.2	22550362
chr9 Vitis vinifera|PN40024|assembly12x.2	23006712
chr10 Vitis vinifera|PN40024|assembly12x.2	23503040
chr11 Vitis vinifera|PN40024|assembly12x.2	20118820
chr12 Vitis vinifera|PN40024|assembly12x.2	24269032
chr13 Vitis vinifera|PN40024|assembly12x.2	29075116
chr14 Vitis vinifera|PN40024|assembly12x.2	30274277
chr15 Vitis vinifera|PN40024|assembly12x.2	20304914
chr16 Vitis vinifera|PN40024|assembly12x.2	23572818
chr17 Vitis vinifera|PN40024|assembly12x.2	18691847
chr18 Vitis vinifera|PN40024|assembly12x.2	34568450
chr19 Vitis vinifera|PN40024|assembly12x.2	24695667
chrUkn Vitis vinifera|PN40024|assembly12x.2	27389308
total	486205130

3.1.5 Reformat sequence headers for `VITVI_PN40024_8x_chroms_URGI`

Launch script:

```
ln -s ../../data/urgi/VV_chr8x.fsa.gz .
echo "../../src/reformat_VV_chr8x.bash" \
  | qsub -cwd -j y -V -N reformat_VV_chr8x -q normal.q
```

Check:

```
zcat VV_chr8x.fsa.gz | wc -l # 8291865
zcat VV_chr8x.fsa.gz | grep -c ">" # 35
zcat VITVI_PN40024_8x_chroms_URGI.fa.gz | wc -l # 8291865
zcat VITVI_PN40024_8x_chroms_URGI.fa.gz | grep -c ">" # 35
diff <(zcat VV_chr8x.fsa.gz) <(zcat VITVI_PN40024_8x_chroms_URGI.fa.gz)
```

Only the headers differ, not the sequences, so everything is fine.

Basic stats:

```
zcat VITVI_PN40024_8x_chroms_URGI.fa.gz | md5sum # 4b6ea1cb4ff189ac587fa269077885b5
```

Length of each sequence:

```
zcat VITVI_PN40024_8x_chroms_URGI.fa.gz \
  | awk 'BEGIN{RS=">"} {split($0,a,"\n");
if(length(a)==0)next;
sum=0; for(i=2;i<=length(a);++i){sum+=length(a[i])};
print a[1]": "sum; sumTot+=sum} END{print sumTot}'
```

header	length (bp)
chr1 CU462738|Vitis vinifera|PN40024|assembly8x|chromosome$_1$	15630816
chr10 CU462747|Vitis vinifera|PN40024|assembly8x|chromosome$_{10}$	9647040
chr10$_{\text{random}}$	2206354
chr11 CU462748|Vitis vinifera|PN40024|assembly8x|chromosome$_{11}$	13936303
chr11$_{\text{random}}$	1958407
chr12 CU462749|Vitis vinifera|PN40024|assembly8x|chromosome$_{12}$	18540817
chr12$_{\text{random}}$	2826407
chr13 CU462750|Vitis vinifera|PN40024|assembly8x|chromosome$_{13}$	15191948
chr13$_{\text{random}}$	1580403
chr14 CU462751|Vitis vinifera|PN40024|assembly8x|chromosome$_{14}$	19480434
chr14$_{\text{random}}$	5432426
chr15 CU462752|Vitis vinifera|PN40024|assembly8x|chromosome$_{15}$	7693613
chr15$_{\text{random}}$	4297576
chr16 CU462753|Vitis vinifera|PN40024|assembly8x|chromosome$_{16}$	8158851
chr16$_{\text{random}}$	4524411
chr17 CU462754|Vitis vinifera|PN40024|assembly8x|chromosome$_{17}$	13059092
chr17$_{\text{random}}$	1763011
chr18 CU462755|Vitis vinifera|PN40024|assembly8x|chromosome$_{18}$	19691255
chr18$_{\text{random}}$	5949186
chr19 CU462756|Vitis vinifera|PN40024|assembly8x|chromosome$_{19}$	14071813
chr19$_{\text{random}}$	1912523
chr1$_{\text{random}}$	5496190
chr2 CU462739|Vitis vinifera|PN40024|assembly8x|chromosome$_2$	17603400
chr2$_{\text{random}}$	60809
chr3 CU462740|Vitis vinifera|PN40024|assembly8x|chromosome$_3$	10186927
chr3$_{\text{random}}$	1343266
chr4 CU462741|Vitis vinifera|PN40024|assembly8x|chromosome$_4$	19293076
chr5 CU462742|Vitis vinifera|PN40024|assembly8x|chromosome$_5$	23428299
chr6 CU462743|Vitis vinifera|PN40024|assembly8x|chromosome$_6$	24148918
chr7 CU462744|Vitis vinifera|PN40024|assembly8x|chromosome$_7$	15233747
chr7$_{\text{random}}$	176143
chr8 CU462745|Vitis vinifera|PN40024|assembly8x|chromosome$_8$	21557227
chr8$_{\text{random}}$	12125
chr9 CU462746|Vitis vinifera|PN40024|assembly8x|chromosome$_9$	16532244
chrUn$_{\text{random}}$	154883714
total	497508771

3.1.6 Format VITVI_PN40024_12x_v0_chroms_URGI for BLASTn

TODO: change Vvin to VITVI

```
../../src/format_Vvin-PN40024-12x-chr_blastn.bash
```

3.1.7 Index VITVI_PN40024_12x_v0_chroms_URGI for BWA

Launch:

```
echo "../../src/bwa_index_VITVI_PN40024_12x_v0_chroms_URGI.bash" \
  | qsub -cwd -j y -V -N bwa_index_VITVI_PN40024_12x_v0_chroms_URGI -q normal.q
```

3.1.8 Index VITVI_PN40024_12x_v2_chroms_URGI for BWA

Launch:

```
echo "../../src/bwa_index_VITVI_PN40024_12x_v2_chroms_URGI.bash" \
  | qsub -cwd -j y -V -N bwa_index_VITVI_PN40024_12x_v2_chroms_URGI -q normal.q
```

3.1.9 Prepare VITVI_PN40024_12x_v2_chroms_URGI for SAMtools and Picard

Make an index as well as a SAM header.

Launch:

```
echo "../../src/samtools-picard_prep_VITVI_PN40024_12x_v2_chroms_URGI.bash" \
  | qsub -cwd -j y -V -N samtools-picard_prep_VITVI_PN40024_12x_v2_chroms_URGI -q
     normal.q
```

3.1.10 Index VITVI_PN40024_12x_v0_chroms_URGI for Bowtie2

Launch:

```
echo "../../src/bowtie2_index_VITVI_PN40024_12x_v0_chroms_URGI.bash" \
  | qsub -cwd -j y -V -N bowtie2_build_VITVI_PN40024_12x_v0_chroms_URGI -q normal.q
```

3.1.11 Index `VITVI_PN40024_12x_v2_chroms_URGI` for Bowtie2

Launch:

```
echo "../../src/bowtie2_index_VITVI_PN40024_12x_v2_chroms_URGI.bash" \
  | qsub -cwd -j y -V -N bowtie2_build_VITVI_PN40024_12x_v2_chroms_URGI -q normal.q
```

3.1.12 Index `VITVI_PN40024_12x_v2_chroms_URGI` for Bowtie2 compatible with Tassel

Tassel requires numbers as chromosome identifiers.

Launch:

```
echo "../../src/bowtie2_index_VITVI_PN40024_12x_v2_chroms_URGI_for_Tassel.bash" \
  | qsub -cwd -j y -V -N bowtie2_build_VITVI_PN40024_12x_v2_chroms_URGI_for_Tassel -
     q normal.q
```

3.1.13 Translate CRIBI annotations from 12X.0 to 12X.2

Requirement: use or write a script taking as input the 12X.0 GFF3 file as well as the 12.0-12.2 AGP file, and returns as output the 12X.2 GFF3 file

The URGI provides the following AGP file: `golden_path_V2_111113_allChr.csv`. Unfortunately, after looking at the official specification of the AGP format, the URGI file doesn't seem to be valid, neither for version 1.1, nor 2.2. After contacting URGI, they told me they were working on it (October 2015).

Another script was developped by G. Sarah, but it suffers from several issues.

TODO: test CrossMap

3.2 NCBI

```
mkdir -p ncbi; cd ncbi/
```

3.2.1 Reformat sequence headers for `VITVI_PN40024_8x_chroms_NCBI`

Launch script:

```
ln -s ../../data/ncbi/ARCHIVE/BUILD.1.1/Assembled_chromosomes/vvi_ref_chr*.fa.gz .
ln -s ../../data/ncbi/ARCHIVE/BUILD.1.1/CHRS/vvi_ref_chrUn.fa.gz
echo "../../src/reformat_chr_NCBI-build-1-1.bash ${i}" \
  | qsub -cwd -j y -V -N reformat_chroms_NCBI-build-1-1 -q normal.q
```

Check:

```
\ls vvi_ref_chr* | while read f; do zcat $f; done | wc -l # 6499942
\ls vvi_ref_chr* | while read f; do zcat $f; done | grep -c ">" # 3343
zcat VITVI_PN40024_8x_chroms_NCBI.fa.gz | wc -l # 6499942
zcat VITVI_PN40024_8x_chroms_NCBI.fa.gz | grep -c ">" # 3343
diff <(\ls -v vvi_ref_chr* | while read f; do zcat $f; done) <(zcat
   VITVI_PN40024_8x_chroms_NCBI.fa.gz)
```

Only the headers differ, not the sequences, so everything is fine.

Basic stats:

```
zcat VITVI_PN40024_8x_chroms_NCBI.fa.gz | md5sum # d8eeff80c824f1b5bd91b4274fddb696
```

3.3 R/Bioconductor

- http://www.bioconductor.org/
- Huber, W. et al. Orchestrating high-throughput genomic analysis with bioconductor. Nature Methods 12, 115-121 (2015). URL http://dx.doi.org/10.1038/nmeth.3252.

TODO: see AnnotationHub

3.3.1 BSgenome IGGP12Xv2 package

http://bioconductor.org/packages/release/bioc/html/BSgenome.html

Retrieve the sequence data from URGI:

```
cd results/
mkdir -p make_BSgenome_IGGP12Xv2
cd make_BSgenome_IGGP12Xv2/
ln -s ../../data/urgi/12Xv2_grapevine_genome_assembly.fa.gz .
```

Split into one chromosome per file (in the headers, discard everything after the first space):

```
zcat 12Xv2_grapevine_genome_assembly.fa.gz | awk 'BEGIN{RS=">"} {if(NF==0)next;
    split($0,a,"\n"); split(a[1],b," "); print b[1]; print ">"b[1] > b[1]".fa"; for(
    i=2;i<length(a);++i){print a[i] >> b[1]".fa"}}'
gzip chr*.fa
```

Prepare the seed file (`IGGP12Xv2_seed.txt`) by hand as indicated in the vignette as well as in the official R manual "Writing R extensions". Following this article, I chose the CC0 license (present in the R list of licenses in `share/licenses/license.db`).

Forge the target package from the seed file:

```
echo "date; echo \"library(BSgenome); forgeBSgenomeDataPkg(\\\"IGGP12Xv2_seed.txt\\\
    ")\" | R --vanilla; date" | qsub -cwd -j y -V -N forge_BSgenome -q normal.q
```

Build the package and check it:

```
echo "date; R CMD build BSgenome.Vvinifera.URGI.IGGP12Xv2; date" | qsub -cwd -j y -V
     -N build_BSgenome -q normal.q
echo "date; R CMD check BSgenome.Vvinifera.URGI.IGGP12Xv2_0.1.tar.gz; date" | qsub -
    cwd -j y -V -N check_BSgenome -q normal.q
```

The target package is now ready to be installed:

```
R CMD INSTALL BSgenome.Vvinifera.URGI.IGGP12Xv2_0.1.tar.gz
```

A.-F. Adam-Blondon (part of IGGP) and other colleagues from INRA gave positive feedback. I hence sent the package to the Bioconductor team. I hope it will soon be available for download via `biocLite()`.

3.3.2 BSgenome IGGP12Xv0 package

`http://bioconductor.org/packages/release/bioc/html/BSgenome.html`

Retrieve the sequence data from URGI:

```
cd results/
mkdir -p make_BSgenome_IGGP12Xv0
cd make_BSgenome_IGGP12Xv0/
ln -s ../../data/urgi/VV_chr12x.fsa.gz .
```

TODO: Prepare the seed file (`IGGP12Xv0_seed.txt`) by editing the one used for `IGGP12Xv2`.

3.3.3 Brief comparison of sequence content between genome files

See script `src/comp_seq_content.R` using Bioconductor.

3.3.4 TxDb IGGP12Xv0 package from NCBI annotations

`http://www.bioconductor.org/packages/release/bioc/html/GenomicFeatures.html`

TODO: use `makeTxDbFromGFF()`