Tutorial for Cortex and Bubbleparse

Richard Leggett
richard.leggett@tgac.ac.uk

April 11, 2013

1 Introduction

This tutorial will show you how to download the Cortex and Bubbleparse sources, how to compile the code, and how to find the SNPs between two different accessions of *Arabidopsis thaliana*.

Because of memory and processing requirements, we recommend that this tutorial is carried out on a cluster instead of a desktop computer.

2 Download and compile sources

1. Download Cortex Con sources from `https://sourceforge.net/projects/cortexassembler/files/cortex_con/`.
2. Download Bubbleparse sources from `https://github.com/richardmleggett/bubbleparse`.
3. Copy `bubbleparse.c` into the `src/util` directory of the Cortex sources.
4. To build Cortex, change into the `cortex_con` build directory (the one containing `Makefile` and the `src` directories) and type:

   ```
   make MAXK=31 cortex_bub
   ```

   Or, if building on Mac OS X:

   ```
   make MAXK=31 MAC=1 cortex_bub
   ```

5. Assuming there are no compilation errors, a file called `cortex_con_31` will appear in the `cortex_con/bin` directory. You might like to copy this to another directory where you store tools.
6. Now to build Bubbleparse, type:

   ```
   make MAXK=31 bubbleparse
   ```

   Or, if building on Mac OS X:

   ```
   make MAXK=31 MAC=1 bubbleparse
   ```

7. A file called `bubbleparse_31` will appear in the `cortex_con/bin` directory.

3 Download *Arabidopsis* reads

1. For this tutorial, we will use the Col-0 and Tsu-1 accessions of *Arabidopsis thaliana*.
2. Create a `data` directory somewhere to store the reads and cortex files.
3. Download two files of Col-0 reads from `http://www.ebi.ac.uk/ena/data/view/SRX000702`. You will end up with two files called `SRR013327.fastq` and `SRR013328.fastq`.
4. Download five files of Tsu-1 reads from `http://www.ebi.ac.uk/ena/data/view/SRX000704`. You will end up with files called `SRR013334.fastq`, `SRR013335.fastq`, `SRR013336.fastq`, `SRR013337.fastq`, `SRR013338.fastq`.

4 Merge reads and make CTX files

1. Within the data directory, create a file of files for the Col-0 reads. Make a text file called `col0reads.txt`:

```
echo "SRR013327.fastq 0" > col0reads.txt
echo "SRR013328.fastq 0" >> col0reads.txt
```

2. Now create a file of files for the Tsu-1 reads:

```
echo "SRR013334.fastq 0" > tsu1reads.txt
echo "SRR013335.fastq 0" >> tsu1reads.txt
echo "SRR013336.fastq 0" >> tsu1reads.txt
echo "SRR013337.fastq 0" >> tsu1reads.txt
echo "SRR013338.fastq 0" >> tsu1reads.txt
```

3. We'll now create a single Cortex CTX file which includes all the Col-0 reads. Use the following command line:

```
cortex_bub_31 -k 21 -n 24 -b 55 -t fastq -c 100 -s 1 -i col0reads.txt
              -o col0.ctx -l col0.log
```

   To execute this command, Cortex will require approximately 32Gb of RAM, so you may need to specify this as a parameter to your cluster's job submission system. On our cluster, this command took around 90 minutes to complete. On successful completion, Cortex will write a CTX file called `col0.ctx` and a log file called `col0.log`.

4. Now for the Tsu-1 reads. Use the following command line:

```
cortex_bub_31 -k 21 -n 24 -b 75 -t fastq -c 100 -s 1 -i tsu1reads.txt
              -o tsu1.ctx -l tsu1.log
```

   To execute this command, Cortex will require approximately 32Gb of RAM. On our cluster, it took around 2 hours to complete.On successful completion, Cortex will write a CTX file called `tsu1.ctx` and a log file called `tsu1.log`.

5 Find bubbles

1. Create a new file of files called `col0tsu1files.txt`:

   ```
   echo "col0.ctx 0" > col0tsu1files.txt
   echo "tsu1.ctx 1" >> col0tsu1files.txt
   ```

2. To run Cortex bubble finding, type the following:

   ```
   cortex_bub_31 -k 21 -n 25 -b 58 -t binary -w 1,200
                 -i col0tsu1files.txt -f col0tsu1output -l col0tsu1.log
   ```

   To execute this command, Cortex will require approximately 48Gb of RAM. On our cluster, it took around 3 hours to complete.

3. Cortex will write two files: `col0tsu1output.fasta` (containing contigs representing paths through bubbles) and `col0tsu1output.coverage` (containing the coverage of those paths).

6 Rank SNPs

1. We need to create an options file for bubbleparse. Create a new file called `bpoptions.txt` containing the following text:

   ```
   EXPECTEDCOVERAGE "0,10,100,0"
   EXPECTEDCOVERAGE "1,10,100,0"
   MINIMUMCONTIGSIZE "100"
   ```

2. Now create a file of files called `allfiles.txt` containing ALL the input files, as follows:

   ```
   SRR013327.fastq 0
   SRR013328.fastq 0
   SRR013334.fastq 1
   SRR013335.fastq 1
   SRR013336.fastq 1
   SRR013337.fastq 1
   SRR013338.fastq 1
   ```

3. Now to run bubbleparse, type:

   ```
   bubbleparse_31 -k 21 -o bpoptions.txt -f col0tsu1output -i allfiles.txt
                  -t table.txt -c table.csv -d bplog.txt -x
   ```

   This should create a ranked table called `table.txt` and the same data represented in comma separated format called `table.csv`.

7 Further information

For further information, please refer to the Cortex and Bubbleparse manuals.

Please report any bugs or problems with the bubble finding options in Cortex, or with bubbleparse, to `richard.leggett@tgac.ac.uk`. General problems with Cortex should be referred to the appropriate authors.