macle2go: Annotate macle Output

Bernhard Haubold
Max-Planck-Institute for Evolutionary Biology, Plön, Germany

June 15, 2018

1 Introduction

Macle computes a complexity measure, the match complexity, by sliding a window across a given genome. A major motivation this computation is to investigate the relationship between sequence complexity and gene function. For instance, we might ask

- Are there more genes in regions of high complexity than expected by chance alone?
- Are genes in high-complexity regions enriched for certain functions?

Macle2go is designed to help answer such questions.

2 Getting Started

Macle2go is written in Go, so assuming a working Go installation, you can get or update the program

```
go get -u github.com/evolbioinf/macle2go
```

and install it

```
go install github.com/evolbioinf/macle2go
```

Now test macle2go

```
macle2go
```

which should list its four subcommands, annotate, enrichment, quantile, and version.

3 Tutorial

We begin by copying the example data files to our working directory

```
cp $GOPATH/github.com/evolbioinf/macle2go/data/*.bz2 .
```

Then we uncompress them

```
bunzip2 *.bz2
```

Now we are ready to work through the three major commands of macle2go, quantile, annotate, and enrichment.

3.1 quantile

The match complexity, C_{M}, varies between 0 and 1. If $C_{\mathrm{M}} = 0$, the region is exactly repeated elsewhere in the genome. Random sequences, on the other hand, have an expected C_{M} of 1. The distribution of C_{M} for random sequences can be modelled by a normal distribution. Using the underlying match length distribution derived by Haubold et al. (2009), we can thus compute the quantiles of C_{M} for random sequences. But first, we list the options of `quantile`

```
macle2go quantile -h
Usage: macle2go quantile options
Example: macle2go quantile -l 2937655681 -g 0.408679 -w 20000 -p 0.05
Options:
        -g <NUM> gc-content
        -l <NUM> genome length
        -w <NUM> window length
        -p <NUM> probability
```

Then run the example command

```
macle2go quantile -l 2937655681 -g 0.408679 -w 20000 -p 0.05
```

where

- `-l` is the sequence length, 2.9 Gb
- `-g` its GC content, 0.41
- `-w` the length of the sliding window, 20 kb
- `-p` the probability mass covered up to the point to be determined, 5%

This gives

```
macle2go quantile -l 2937655681 -g 0.408679 -w 20000 -p 0.05
#SeqLen WinLen P Q F(Q)
2.937655681e+09 20000 0.05 0.9967710269575726 52.53776662435675
```

where the sequence length, window length, and probability are repeated from the input. `Q` (0.9968) is the quantile, and `F(Q)` (52.54) the value of the normal distribution at this point.

3.2 annote

Again, we start by listing the options:

```
macle2go annotate -h
Usage: macle2go annotate options [inputFiles]
Example macle2go annotate -r hsRefGene.txt -c 0.9968 -w 20000 hs_20k.mac
Options:
        -r <FILE> refGene file, e. g. http://hgdownload.soe.ucsc.edu/
           goldenPath/hg38/database/refGene.txt.gz
        -w <NUM> window length
        -c <NUM>  minimum complexity
        [-C <NUM> maximum complexity; default: no upper limit]
        [-I <NUM> iterations; default: 10000]
        [-s <NUM> seed for random number generator; default: system-
           generated]
        [-u <NUM> upstream promoter region; default: 1000]
        [-d <NUM> downstream promoter region; default: 1000]
        [-G consider whole genes; default: promoter]
```

The file `hs_20k.mac` contains complexity data calculated by `macle` for the complete human genome with 20 kb sliding windows. List the first three lines:

```
head -n 3 hs_20k.mac
chr1 10000 -1.0000
chr1 12000 -1.0000
chr1 14000 -1.0000
```

The first column gives the chromosome name, followed by the midpoint of the sliding window and the C_{M} value. If $C_{\mathrm{M}} = -1$, as in our example, not enough nucleotides were sequenced in that window to compute a meaningful C_{M}-value.

Our aim is now to extract all regions with a complexity equivalent to that of a random sequence. We have just computed the cutoff value for this using `quantile`, so `annotate` should extract regions with $C_{\mathrm{M}} \geq 0.9968$. Each extracted region is annotated with the list of genes with intersecting promoters. A promoter is defined as an interval around the transcription start site (TSS). In `annotate` the default promoter is $\mathrm{TSS} \pm 1\mathrm{kb}$, but this can be adjusted using the options `-u` and `-d`. The TSS of all human genes are listed in the file `hsRefGene.txt`, the latest version of which is available from

`http://hgdownload.soe.ucsc.edu/goldenPath/hg38/database/refGene.txt.gz`

We can now run `annotate` and list the first 10 lines of its output:

```
macle2go annotate -r hsRefGene.txt -w 20000 -c 0.9968 hs_20k.mac | head
# W I O E O/E P
# 700 181 339 164.58 2.06 -0.0001
# Chr Start End Len C_M Sym
chr1 1320001 1346000 26000 0.9974 CPTP INTS11 MIR6808 TAS1R3
chr1 2298001 2324000 26000 0.9999
chr1 2496001 2536000 40000 1.0045 HES5 PANK4
chr1 3058001 3088000 30000 1.0094 LINC00982 PRDM16
chr1 3116001 3142000 26000 1.0004 MIR4251
chr1 3216001 3238000 22000 0.9972
chr1 8348001 8368000 20000 0.9968
```

The first two lines summarize the result:

- 700 windows had $C_{\mathrm{M}} \geq 0.9968$ (`W`).
- Overlapping windows were merged to yield 181 intervals (`I`).
- 339 genes were observed to intersect the 181 intervals (`O`).
- 164.58 genes were expected (`E`) by repeatedly sampling 700 random windows. By default, the re-sampling is repeated 10^4 times, which can be changed using the `-I` option.
- The ratio of observed to expected genes is 2.06.
- The hypothesis that `E` $\geq$ `O` is tested by computing the frequency of observing random samples containing more than 339 genes. The negative value of `P` is meant to be read as "less than", in our example $< 10^{-4}$.

The summary is followed by the 181 intervals identified. Each interval is described in one line consisting of six columns:

1. Chromosome
2. Start position
3. End position
4. Length
5. C_{M}
6. Genes, possibly none, whose promoters intersect the region

3.3 enrichment

Next we investigate whether the sample of 339 genes just discovered is enriched for particular functions. For this we need to link the names of genes to functional categories. The file `gene2go`, which was obtained from

```
ftp://ftp.ncbi.nih.gov/gene/DATA/gene2go.gz
```

lists this information:

```
head -n 2 gene2go
#tax_id GeneID GO_ID Evidence Qualifier GO_term PubMed Category
3702 814629 GO:0005634 ISM -nucleus -Component
```

"GO" stands for "gene ontology", a system for describing gene functions (The Gene Ontology Consortium, 2000). Notice that genes are denoted by `GeneID`, rather than symbol. So we need a second file to connect symbols to `GeneID`. This is the purpose of `Homo_sapiens.gene_info`, obtained from

```
ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia/Homo_sapiens.gene_info.gz
```

It contains the following 16 columns:

```
head -n 1 Homo_sapiens.gene_info | tr '\t' '\n' | cat -n
     1 #tax_id
     2 GeneID
     3 Symbol
     4 LocusTag
     5 Synonyms
     6 dbXrefs
     7 chromosome
     8 map_location
     9 description
    10 type_of_gene
    11 Symbol_from_nomenclature_authority
    12 Full_name_from_nomenclature_authority
    13 Nomenclature_status
    14 Other_designations
    15 Modification_date
    16 Feature_type
```

So columns 2 and 3 connect `GeneID` to `Symbol`:

```
cut -f 2,3 Homo_sapiens.gene_info | head -n 3
GeneID Symbol
1 A1BG
2 A2M
```

However, we need not worry about this as `enrichment` automatically connects symbols to GO-categories via `GeneID`. We begin by listing its options:

```
macle2go enrichment -h
Usage: macle2go enrichment options [inputFiles]
Example: macle2go enrichment -i Homo_sapiens.gene_info -g gene2go -r hsRefGene.txt -c 0.
Options:
-r <FILE> refGene file, e. g. \
           http://hgdownload.soe.ucsc.edu/goldenPath/hg38/database/refGene.txt.gz
-i <FILE> gene-info file, e. g. \
           ftp://ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia/Homo_sapiens.gene_info.gz
-g <FILE> gene2go file, e. g. ftp://ftp.ncbi.nih.gov/gene/DATA/gene2go.gz
-c <NUM>  minimum complexity
```

```
-w <NUM>  window length
[-C <NUM>  maximum complexity; default: no upper limit]
[-I <NUM> iterations; default: 10000]
[-m <NUM>  minimum number of genes per GO-category; default: 10]
[-s <NUM> seed for random number generator; default: system-generated]
[-u <NUM> upstream promoter region; default: 1000]
[-d <NUM> downstream promoter region; default: 1000]
[-G analyze whole genes; default: promoter]
```

Next, we run the example command and save the result to a file:

```
macle2go enrichment -i Homo_sapiens.gene_info -g gene2go -r
   hsRefGene.txt -c 0.9968 -w 20000 hs_20k.mac > enr.txt
```

We take a look at the result

```
head -n 2 enr.txt
GO:0000122 48 9.97 4.82 -1.00e-04
    negative_regulation_of_transcription_by_RNA_polymerase_II
GO:0000977 28 3.17 8.83 -1.00e-04 RNA_polymerase_II_regulatory_region_sequence-
    specific_DNA_binding
```

It consists of six columns:

1. GO accession
2. Observed number of genes, O
3. Expected number of genes, E
4. Ratio of observed to expected genes
5. P-value of the null hypothesis that $E \geq O$, which is $< 10^{-4}$ in the two examples shown
6. Description of the GO-term

As a final step in this tutorial we extract all categories with $P < 10^{-4}$ and sort them in reverse by the ratio of O/E:

```
awk '$5<0' enr.txt | sort -n -k 4 -r | head -n 5
GO:0009954 11 0.29 38.50 -1.00e-04 proximal/distal_pattern_formation
GO:0009952 31 0.97 31.89 -1.00e-04 anterior/posterior_pattern_specification
GO:0048704 14 0.48 29.46 -1.00e-04 embryonic_skeletal_system_morphogenesis
GO:0042472 11 0.70 15.61 -1.00e-04 inner_ear_morphogenesis
GO:0042475 10 0.68 14.77 -1.00e-04 odontogenesis_of_dentin-containing_tooth
```

References

B. Haubold, P. Pfaffelhuber, M. Domazet-Lošo, and T. Wiehe. Estimating mutation distances from unaligned genomes. *Journal of Computational Biology*, 16:1487–1500, 2009.

The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.