Note S1   Tool availability and implementation

Smash++ is implemented in `C++` language and is available at [1]. This tool is able to find and visualize rearrangements in sequences, FASTA and FASTQ files; although, it is highly recommended to use sequences as input. In the following sections, we describe installing and running Smash++.

S1.1   Install

In order to install Smash++ on Linux, run the following commands:

```
1  git clone https://github.com/smortezah/smashpp.git
2  cd smashpp
3  cmake .
4  make
```

S1.2   Run

A reference file and a target file are clearly mandatory to run Smash++ (without visualization). Running

```
1  ./smashpp
```

provides the following guide:

```
1  SYNOPSIS
2    ./smashpp  OPTIONS...  -r REF-FILE  -t TAR-FILE
3
4  SAMPLE
5
6  DESCRIPTION
7    Mandatory arguments
8    -r,  --ref FILE           reference file (Seq/Fasta/Fastq)
9    -t,  --tar FILE           target file   (Seq/Fasta/Fastq)
10
11   Options
12   -v,  --verbose            more information
13   -l,  --level INT          level of compression: [0, 5]
14   -m,  --min   INT          min segment size: [1, 4294967295]
15   -nr, --no-redun           do NOT compute self complexity
16   -e,  --ent-n FLOAT        Entropy of 'N's: [0.0, 100.0]
17   -n,  --nthr  INT          number of threads: [1, 8]
18   -fs, --filter-scale S|M|L scale of the filter:
19                             {S|small, M|medium, L|large}
20   -w,  --wsize INT          window size: [1, 4294967295]
21   -wt, --wtype INT/STRING   type of windowing function:
22                             {0|rectangular, 1|hamming, 2|hann,
23                             3|blackman, 4|triangular, 5|welch,
24                             6|sine, 7|nuttall}
25   -d,  --step   INT         sampling steps
26   -th, --thresh FLOAT       threshold: [0.0, 20.0]
27   -sb, --save-seq           save sequence (input: Fasta/Fastq)
28   -sp, --save-profile       save profile (*.prf)
29   -sf, --save-filter        save filtered file (*.fil)
30   -ss, --save-segment       save segmented files (*-s_i)
31   -sa, --save-all           save profile, filetered and
32                             segmented files
33   -h,  --help               usage guide
34   -rm, --ref-model  k,[w,d,]ir,a,g/t,ir,a,g:...
35   -tm, --tar-model  k,[w,d,]ir,a,g/t,ir,a,g:...
36                             parameters of models
37                     (INT) k:  context size
38                     (INT) w:  width of sketch in log2 form,
39                           e.g., set 10 for w=2^10=1024
40                     (INT) d:  depth of sketch
41                     (INT) ir: inverted repeat: {0, 1, 2}
```

```
42                                              0: regular (not inverted)
43                                              1: inverted, solely
44                                              2: both regular and inverted
45                         (FLOAT) a:   estimator
46                         (FLOAT) g:   forgetting factor: [0.0, 1.0)
47                           (INT) t:   threshold (no. substitutions)
```

The arguments "-r" and "-t" are used to specify the reference and the target, respectively, which are highly recommended to have short names. Level of compression, that is an integer between 0 and 5, can be determined with "-l". By setting "-m" to an integer value, only those regions in the reference file that are bigger than that value can be considered for compression. Triggering "-nr" makes the tool not to perform the reference-free compression (self-complexity computation) part.

In implementation of the reference-based compression, we have replaced 'N' bases in the references and the targets with 'A's and 'T's, respectively. On reference-free compression, they are replaced with 'A's, in both references and targets. If a user tends to replace 'N' bases in a sequence with a normal distribution of 'A', 'C', 'G' and 'T's, he/she can employ GOOSE toolkit [2]. Note that we have set the entropy of 'N's to 2.0, by default, but it is possible for the user to set them to another value of interest, by "-e" option.

Building different finite-context models can be done in the multi-threaded fashion, setting "-n" to an integer. To find similar regions in the reference and the target, information profile (obtained by compression) needs to be filtered, of which the scale can be set as S (small), M (medium) or L (large). Size of the window and type of the windowing function, described in **??**, can be set by "-w" and "-wt" options, respectively. Instead of considering the complete profile information, the user is able to make samples of it by steps of which size can be determined by "-d". For the purpose of segmenting the filtered information profile, the average entropy of reference-based compression is used as the threshold, by default. However, this threshold can be altered by "-th" option.

Smash++ accepts FASTA and FASTQ files as input, in addition to sequences. In these cases, the input files are converted to sequences and then processed further. It is possible to save these sequences by "-sb" option.

After obtaining the information profile, Smash++ filters it and then removes it, by default. However, it is possible to save the profile by "-sp" option. The same thing happens to the filtered file, i.e., it is segmented and then is removed. But, the user can use "-sf" to save the filtered file. Also, the segmented files can be saved using "-ss". The user can save all the information profile, filtered and segmented files, by triggering "-sa" option.

For the purpose of compression, either reference-based or reference-free, it is recommended to use "-l" option, since it configures the models automatically. However, using "-rm" and "-tm", the user would be able to manually configure the reference model, for reference-based compression, and the target model, for reference-free compression. Parameters of the models are described in detail in section **??**.

Running Smash++ (without visualization), positions of the similar regions in the reference and the target, and also complexity of the regions is saved in a "*.pos" file. To visualize this file, one can type

```
1   ./smashpp -viz
```

which gives

```
1   SYNOPSIS
2     ./smashpp -viz  OPTIONS...  -o SVG-FILE  POS-FILE
3
4   SAMPLE
5
6   DESCRIPTION
7     Mandatory arguments:
8     POS-FILE                    positions file, generated by
9                                 Smash++ tool (*.pos)
10
11    Options:
12    -v,  --verbose              more information
```

```
13    -o ,   --out SVG -FILE    output image name (*. svg)
14    -rn , --ref -name STRING  reference name shown on output. If name
15                              has space , use "s, e.g. "Seq label ".
16                              Default: name in header of position file.
17    -tn , --tar -name STRING  target name shown on output
18    -vv , --vertical          vertical view
19    -nn , --no -nrc           do NOT show normalized
20                              relative compression (NRC)
21    -nr , --no -redun         do NOT show self complexity
22    -ni , --no -inv           do NOT show inverse maps
23    -ng , --no -reg           do NOT show regular maps
24    -l ,   --link     INT     type of the link between maps: [1, 6]
25    -c ,   --color    INT     color mode: [0, 2]
26    -p ,   --opacity  FLOAT   opacity: [0.0 , 1.0]
27    -w ,   --width    INT     width of the sequence: [15, 100]
28    -s ,   --space    INT     space between sequences: [15, 200]
29    -f ,   --mult     INT     multiplication factor for
30                              color ID: [1, 255]
31    -b ,   --begin    INT     beginning of color ID: [0, 255]
32    -rt , --ref -tick INT     reference tick: [1, 4294967295]
33    -tt , --tar -tick INT     target tick: [1, 4294967295]
34    -th , --tick -human 0|1   tick human readable: 0=false , 1=true
35    -m ,   --min      INT     minimum block size: [1, 4294967295]
36    -h ,   --help             usage guide
```

The output of Smash++ visualizer is an "SVG" file, which its name is determined by "-o" option. By default, it is named "map.svg". Names of the reference and the target, which are going to be printed on the output image, can be altered by "-rn" and "-tn", respectively. They are by default the names written in the positions file. To have a vertical view of the image, instead of the default horizontal view, one can use "-vv" trigger.

Smash++ performs reference-based and reference-free compressions to calculate the normalized relative compression (NRC) and redundancy (self complexity), respectively. If the user is not interested in showing them, he/she can turn them off by "-nn" and "-nr" triggers. In addition, Smash++ considers both regular and reverse complement maps by default in its calculations. Triggering "-ni" and "-ng" will stop showing inverted and regular maps, respectively.

Options "-l", "-c", "-p", "-w", "-s", "-f" and "-b" can be used to change the appearance of the image. Assigning integers to "-rt" and "-tt" options will change the tick sizes of the reference and the target, respectively. Smash++ prints the sizes on axes in human readable format, e.g., 1K, 2M, etc. However, it can be triggered by "-th" option. Note that, here, 1K is equivalent to 1000 and not 1024. Finally, by setting "-m" to an integer value, only the regions that are bigger than that value will be illustrated.

S1.3   Example

This section guides, step-by-step, employing Smash++ to find and visualize rearrangements in a sample genomic data.

Install Smash++ and provide the required files

First, we install Smash++:

```
1  git clone https :// github.com/smortezah/smashpp.git
2  cd smashpp
3  cmake .
4  make
```

Then, we copy `smashpp` binary file into `example/` directory and go to that directory:

```
1  cp smashpp example/
2  cd example/
```

In this directory, we have provided a 1000 byte reference sequence, `ref`, and a 1000 byte target sequence, `tar`.
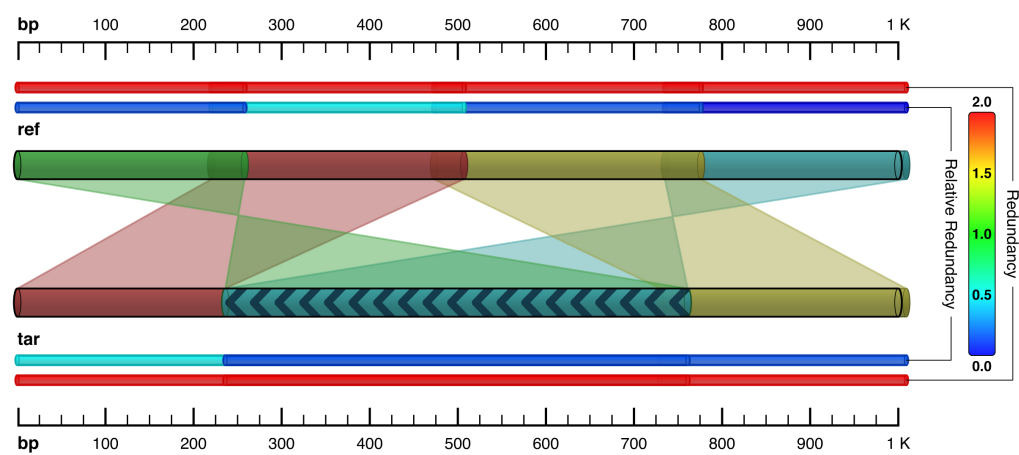
The result is shown in Fig. S1.

Fig. S1. The result of running Smash++ on ...

References

[1] M. Hosseini, D. Pratas, and A. J. Pinho. Smash++. [Online]. Available: https://github.com/smortezah/smashpp

[2] D. Pratas. Goose. [Online]. Available: https://github.com/pratas/goose