Note S1 Tool availability and implementation

Smash++ is implemented in `C++` language and is publicly available at [1], under GNU GPL v3 license. This tool is able to find and visualize rearrangements in a pair of DNA sequences. It is recommended to use as input bare sequences, i.e., without header or quality scores, although, FASTA and FASTQ formats are also supported. In the following sections, we describe installing and running the Smash++ tool.

S1.1 Run Smash++ visualizer

The position file obtained by Smash++ can be visualized by

```
./smashpp -viz
```

The visualizer provides various options that are described in Table S1. The commands for running Smash++ visualizer are of the form

```
./smashpp -viz [OPTIONS]  -o <SVG-FILE>  <POS-FILE>
```

Table S1. Options provided by Smash++ visualizer interface.

Flag	Input	Description
Required		
	*.pos file	Position file, generated by Smash++ tool. It can be redirected to Smash++ by standard input ("stdin").
Optional		
`-o`	*.svg file Default: map.svg	Output image name.
`-rn`	String	Reference name shown on output.
`-tn`	String	Target name shown on output.
	Default: names in position file's header	If it has some spaces, use double quotes, e.g. "Seq label".
`-l`	Integer: $[1, 6]$ Default: 1	Type of the link between similar regions.
`-c`	Integer: $[0, 1]$ Default: 0	Color mode.
`-p`	Float: $[0.0, 1.0]$ Default: 0.9	Opacity.
`-w`	Integer: $[8, 100]$ Default: 10	Width of the sequence.
`-s`	Integer: $[5, 200]$ Default: 40	Space between sequences.
`-tc`	Integer: $[1, 255]$ Default: 40	Total number of colors in the map, which is automatically chosen by default.
`-rt`	Integer: $[1, 2^{32} - 1]$	Reference tick size.
`-tt`	Integer: $[1, 2^{32} - 1]$	Target tick size.
`-th`	Integer: $\{0, 1\}$ Default: 1	Tick human readable: 0=false, 1=true. If it is true, the sizes on axes are printed in the format 1K, 2M, etc. Note that here, 1K is equivalent to 1000 and not 1024, and so on.
`-m`	Integer: $[1, 2^{32} - 1]$ Default: 1	Minimum block size. Only the regions with greater sizes than this value will be illustrated.
`-vv`	—	Vertical view of the output image.
`-nn`	—	Do not show normalized relative compression (NRC).
`-nr`	—	Do not show redundancy (self complexity).

		Smash++ performs reference-based and reference-free compressions to calculate the NRC and redundancy, respectively. If a user does not tend to show them, he/she can turn them off by “-nn” and “-nr” triggers.
`-ni` `-ng`	— —	Do not show inverse maps. Do not show regular (not inverse) maps. Smash++ considers by default both regular and reverse complement maps in its calculations.
`-h`	—	Usage guide.
`-v`	—	More information (verbose).
`--version`		Show version.

S1.2 Example

This section guides step-by-step employing Smash++ to find and visualize rearrangements in a sample genomic data. Note that the commands can be run on Linux and macOS, however, they are similar in Windows.

Install Smash++ and provide the required files

First, install Smash++:

```
git clone https://github.com/smortezah/smashpp.git
cd smashpp
./install.sh
```

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

```
cp smashpp example/
cd example/
```

There is a 1000 byte reference sequence, named `refs`, and a 1000 byte target sequence, named `tars`, in this directory. Running

```
./smashpp -r refs -t tars -f 45 -l 3
./smashpp -viz -p 1 -s 50 -w 15 refs.tars.pos
```

results in Fig. **??**, which has been saved as “map.svg”.

References

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