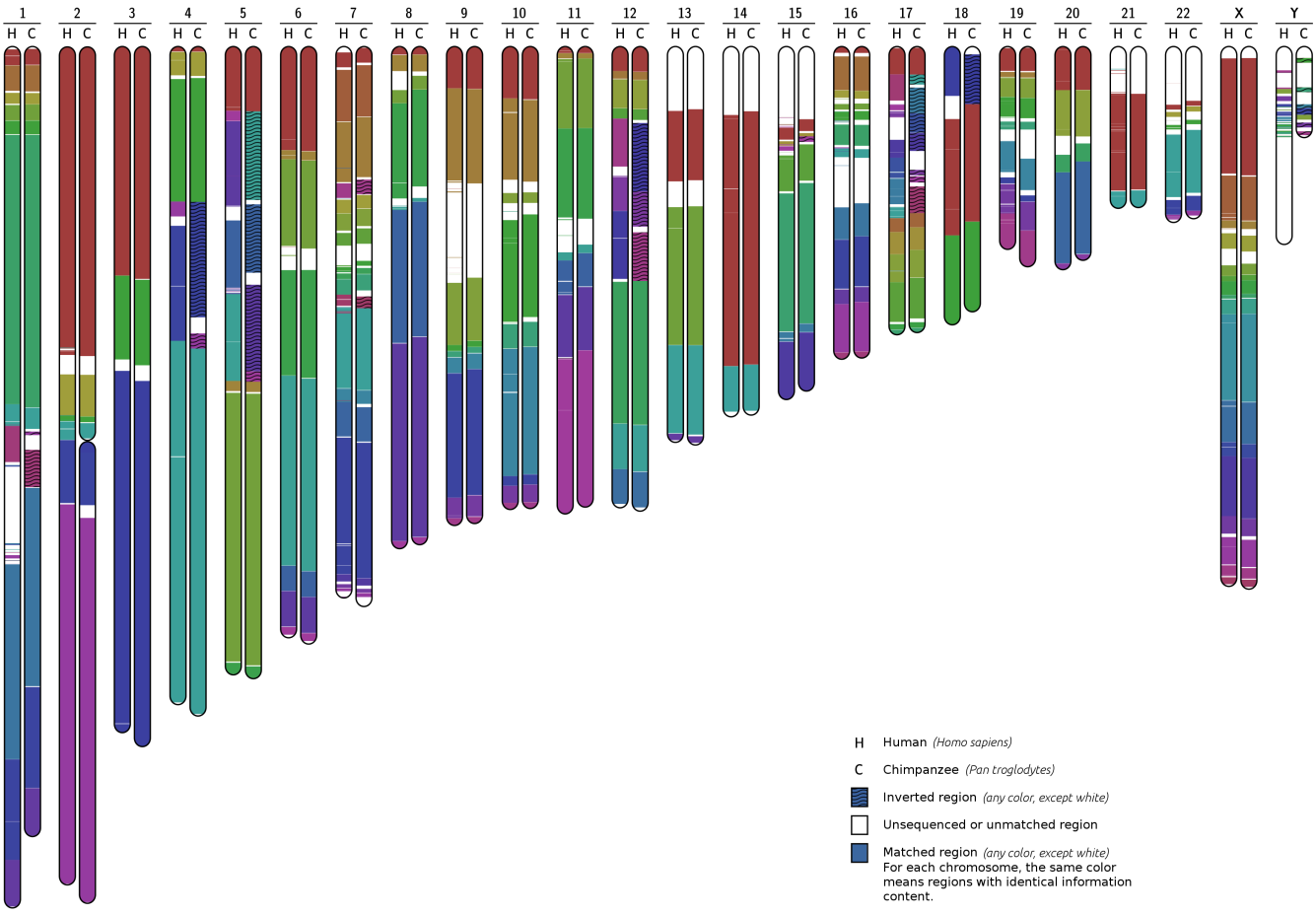
# Manual of SMASH

***Diogo Pratas, Raquel M. Silva, Armando J. Pinho & Paulo J. S. G. Ferreira***

***pratas@ua.pt , raquelsilva@ua.pt , ap@ua.pt , pjf@ua.pt***

***IEETA/DETI University of Aveiro, Portugal***

**Smash is a completely alignment-free method to find and visualise rearrangements between pairs of DNA sequences**. The detection is based on **conditional exclusive compression**, namely using a **FCM**, also known as Markov model, of high context order (typically 20). The method has been approached with a **tool** (also called Smash). For visualization, Smash outputs a **SVG image**, with an **ideogram** output architecture, where the patterns are represented with several **HSV** values (only value varies). The following image, illustrating the information maps between human and chimpanzee for the several chromosomes, depicts an example:



# INSTALLATION

We provide a binary for each 64 bits operating systems (Linux, OSX, Windows). However, for other purposes, such as source code compilation, use the following installation instructions.

Cmake is needed for installation (http://www.cmake.org/). You can download it directly from http://www.cmake.org/cmake/resources/software.html or use an appropriate packet manager. In the following instructions we show the procedure to install, compile and create the information map between human and orangutan chromosome 20:

## STEP 1

Download, install smash and resolve conflicts.

### Linux

```
sudo apt-get install cmake
wget https://github.com/pratas/smash/archive/master.zip
unzip master.zip
cd smash-master
cmake .
make
```

### OS X

Install brew:

```
ruby -e "$(curl -fsSL https://raw.github.com/Homebrew/homebrew/go/install)"
```

only if you do not have it. After type:

```
brew install cmake
brew install wget
brew install gcc48
wget https://github.com/pratas/smash/archive/master.zip
unzip master.zip
cd smash-master
cmake .
make
```

With some versions you might need to create a link to cc or gcc (after the *brew install gcc48* command), namely

```
sudo mv /usr/bin/gcc /usr/bin/gcc-old   # gcc backup
sudo mv /usr/bin/cc /usr/bin/cc-old     # cc backup
sudo ln -s /usr/bin/gcc-4.8 /usr/bin/gcc
sudo ln -s /usr/bin/gcc-4.8 /usr/bin/cc
```

In some versions, the gcc48 is installed over /usr/local/bin, therefore you might need to substitute the last two commands by the following two:

```
sudo ln -s /usr/local/bin/gcc-4.8 /usr/bin/gcc
sudo ln -s /usr/local/bin/gcc-4.8 /usr/bin/cc
```

### Windows

In windows use cygwin (https://www.cygwin.com/) and make sure that it is included in the installation: cmake, make, zcat, unzip, wget, tr, grep (and any dependencies). If you install the complete cygwin packet then all these will be installed. After, all steps will be the same as in Linux.

## Step 2

Download the sequences [the links might change over time].

#### Linux and OS X

```
wget
ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/Assembled_chromosomes/seq/hs_ref_GRCh
38_chr20.fa.gz
wget
ftp://ftp.ncbi.nlm.nih.gov/genomes/Pongo_abelii/Assembled_chromosomes/seq/pab_ref_
P_pygmaeus_2.0.2_chr20.fa.gz
```

## Step 3

Unzip, exclude headers and filter content.

#### Linux

```
zcat hs_ref_GRCh38_chr20.fa.gz | grep -v ">" | tr -d -c "ACGTN" > HS20
zcat pab_ref_P_pygmaeus_2.0.2_chr20.fa.gz | grep -v ">" | tr -d -c "ACGTN" > PA20
```

#### OS X

```
gzcat hs_ref_GRCh38_chr20.fa.gz | grep -v ">" | tr -d -c "ACGTN" > HS20
gzcat pab_ref_P_pygmaeus_2.0.2_chr20.fa.gz | grep -v ">" | tr -d -c "ACGTN" > PA20
```

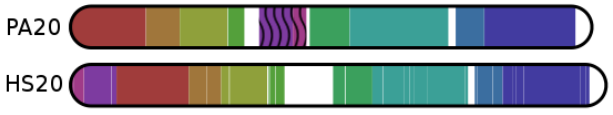
## Step 4

Run Smash.

#### Linux and OS X

```
./smash -v -c 20 -t 1.5 HS20 PA20
```

This step outputs a SVG image using the respective map under the name: HS20PA20.svg (for custom name use option: -o ANYNAME.svg). The respective information map is the following (transformed in a png image and rotated 90 degrees to left):



# USAGE

The Smash program have many options in the interface because there are a wide variety of parameters that can be defined by the user. However, for the detection of the arrangements only two are critical, namely context and threshold. Mathematical information about these parameters can be found in the paper.

To see the possible options type

```
./smash
```

or

```
./smash -h
```

These will print the following options:

```
Usage: smash [OPTIONS]... [FILE] [FILE]

 -v                  verbose mode
 -f                  force (be sure!)

 -c  [context]       context order (DEF: 20)
 -t  [threshold]     threshold [0.0,2.0] (DEF: 1.5)

 -m  [mSize]         minimum block size (DEF: 1000000)
 -i                  show only inverted repeats
 -r  [ratio]         image size ratio (DEF: 1000000)
 -a  [alpha]         alpha estimator (DEF: 1000)
 -s  [seed]          seed for random 'N'
 -w  [wSize]         window size
 -wt [wType]         window type [0|1|2|3] (DEF: 0)
 -d  [dSize]         sub-sample (DEF: 10000)
 -wi [width]         sequence width (DEF: 25)

 -o  [outFile]       output svg plot file

 [refFile]           reference file
 [tarFile]           target file
```

By default, Smash has many parameters assigned in order to avoid the estimation, enabling only to set both reference and target files. However, these defaults are only estimated to detect rearrangements in primates. Therefore, for other purposes you might need to adjust context and threshold parameters. Moreover, for custom image maps you might also need to set other parameters, such as width. Only [refFile] and [TarFile] are mandatory.

### Options meaning

| Parameters | Meaning |
|---|---|
| -v | It will print progress information such as positions of the patterns, times, etc. |
| -f | It will force to write over files already created. |
| -c [context] | Size of the FCM (Markov) context order (interval [1;28]). Contexts above 14 will be handled with a hash-table, where the implementation is approximately linear in memory |

| Parameters | Meaning |
|---|---|
| | relatively to the size of the sequence. When the sequence is very fragmented, or the species are somehow distant, or the sequencing/assembly process has low quality this value show not be very high. |
| -t [threshold] | It will be used to segment the high from the low regions of information content (interval [0;2]). For distant species this value might be slightly below 2 (such as 1.9). |
| -m [mSize] | Minimum size of the block considered as a valid patters after each segmentation process. Values below 1 Million for primate chromosomes might emerge excessive valid patterns. However for other purposes, such as gene scale analysis, this value should be set almost to 1. |
| -i | It will detect and show only the information map regarding to inversions. |
| -r [ratio] | Sets the ratio size of the image. Currently is fixed to 1000000 which is an estimated value to the medium of the primate chromosomes sizes relatively to the medium of the screen resolution. This parameter is not automatically adapted since a fixed value will bring different size chromosomes to the same scale. Nevertheless, to use it in small sequences, namely bacterial genomes, this parameter might be adjusted. |
| -a [alpha] | Probabilities estimator. This value relates a linear interpolation between the maximum likelihood estimator and the uniform distribution. This also shows that when the total number of events is large, the estimator behaves as a maximum likelihood estimator. Default value is set to 1000. |
| -s [seed] | This is a parameter to the pseudo-random generation function. Different seed values ensure different generated values. |
| -w [wSize] | The window size among with the sub-sampling is calculated automatically, nevertheless this value might be adjusted for special needs. |
| -wt [wType] | Window filtering type. Types: 0, 1, 2 or 3. Type 0 stands for Hamming, 1 for Hann, 2 for Blackman, while 3 represents a rectangular window. |
| -d [dSize] | Sub-sampling value. This value among with the window size is calculated automatically. Nevertheless, for special purposes this value might be adjusted. |
| -wi [width] | Thickness of the image for each sequence. Default value is set to 25. |
| -o [outFile] | The output SVG image filename. The default uses the concatenation of reference with the target filenames (adding the "svg" extension). Beware: if the files are not in the working directory this might have problems due to several types of characters (such as '/'). |
| [refFile] | The reference filename. |
| [tarFile] | The target filename. |

# CITATION

On using this software/method please cite:

Diogo Pratas, Raquel M. Silva, Armando J. Pinho, Paulo J. S. G. Ferreira. An alignment-free method to find and visualize genomic rearrangements, 2014. DOI: doi-to-appear

# ISSUES

For any issue let us know at issues link.

# LICENSE