Quick start

February 28, 2011

1 Installation

1.1 Requirements

Segtor requires the Perl interpreter, a C++ compiler and the following Perl modules:

- Cwd
- Data::Dumper
- File::stat
- File::Basename
- Getopt::Std
- List::Util
- LWP
- POSIX
- Storable
- Time::Local
- Time::HiRes

1.2 Installation

To install this program type the following:

```
cd seq-substring
gcc -O3 -o seq-substring seq-substring.c
cd ..
chmod u+x Segtor/main.pl
```

2 Running Segtor

Segtor requires two files to run:

1. An index detailing the available files and chromosome names for the given genome assembly.

2. A compressed segment tree for the given database and genome assembly.

The compressed segment trees are loaded in the RAM to annotate the coordinates. Please remember that there are 4 types of indices:

1. Index files with a segment trees without the genomic sequence (with a . treedat suffix) for annotating coordinates and intervals.

2. Index files with a segment trees with the genomic sequence(with a . treesnvdat suffix) for annotating SNV, insertions, deletions and translocations. Larger than the previous one.

3. Index files with a sorted array of the tss (with a .tssdat suffix) for detecting the closest TSS

4. Index files with a hash of the dbSNP (with a .dbsnp) suffix for detecting existing SNP in the dbSNP

The user has 2 options to get the files to run Segtor, either :

- Use the ones available on our website. See Section 2.1.

- Build their own. This builds both types of indices automatically. See section 2.2.

The following sections provides instructions for both. Bear in mind that Segtor requires a directory on the local disk to store the index files. By default, Segtor stores its data in the user's home directory in .segtor/ but it can change it using -c option. Some input samples can be found in the Segtor/sampleData/ directory.

2.1 Using a pre-built index file

As of now, we only provide indices that work on a Intel x86_64 architecture. If you use a different processor, please refer to section to learn how to build your own indices for a different CPU architecture. After having downloaded the files, please remember to unzip them. Segtor requires 2 files to run:

1. A file containing the list of files for the given assembly and the chromosomes contained in the files. This file is available on Segtor's website.

2. A segment tree index file. Pre-build index files for a given genome assemblies can be downloaded from Segtor's website.

Segtor needs a directory to store the indices. By default, Segtor will look in the home directory and we recommend to store the files there. However, if the user has size quotas or wish to share Segtor between many users of a system, the indices can be stored elsewhere but must be specified with the -c option. To create Segtor's directory in the default location, the user foobar with /home/foobar/ as home directory will create the following directory:

```
/home/foobar/.segtor/
```

In your that directory, create a directory using the nomenclature used on the UCSC Genome Browser for the given species you wish to use and remember to use the correct case. (e.g. hg19 for the Feb. 2009 human genome). So the user "foobar" will create the directory:

```
/home/foobar/.segtor/hg19/
```

Remember that the base location of this directory can be changed using the -c option.
Within that directory, create a directory "chromosomes" and another called "structures". Store the chromosome index file (normally index.txt) in:

```
/home/foobar/.segtor/hg19/chromosomes/
```

and store the segment tree index (the file ending either in .treedat, .tssdat, .treesnpdat) in:

```
/home/foobar/.segtor/hg19/structures/
```

Once you have both files in the proper directories, you should be able to run Segtor. Jump to Section 2.3 for instructions.

2.2 Building your own index file

This section is for user who seeks to build their own index files for a given species and database. This is the best option to use the most recent gene information or a new genome assembly. Before you proceed, make sure that:

1. Make sure you are connected to the internet and the velocity is suitable for downloading data.

2. The genome assembly you seek is available on the UCSC Genome Browser FTP site and you have the correct species code (case sensitive). Verify by going to: ftp://hgdownload.cse.ucsc.edu/goldenPath/

3. The gene database genome assembly you seek is available for the genome assembly your are planning to use on the UCSC Genome Browser FTP site. Verify by going to: ftp://hgdownload.cse.ucsc.edu/goldenPath/

4. Segtor needs the genomic sequence and will download the entire genome plus the database file you seek to use. Please make sure you enough disk space. By default Segtor stores its data in your home directory in .segtor/ but you can change it using -c option.

Alternatively, if you already have existing psl or bed files describing the features you want to use as custom databases, please refer to section 2.2.2.

2.2.1 Examples of creating indices

To create an index for Human hg19 with ests for coordinates

```
./main.pl -s hg19 -m 1 -d ests -o
```

To create an index for Mouse mm9 with knowngene for intervals and do not promp for download

```
./main.pl -s mm9 -m 2 -d knowngene -o -y
```

To create an index for Rat rn4 with ensembl for SNVs using proxy http://123.58.13.21:34

```
./main.pl -s rn4 -m 3 -d ensembl -o -p http://123.58.13.21:34
```

To create an index for Human hg18 with ensembl for SNVs and build an index for dbSNP and use /foo/bar to store the data

```
./main.pl -s hg18 -m 3 -d ensembl -o -b -c /foo/bar
```

To create an index for Chimp panTro2 with refSeq for detecting the closest TSS

```
./main.pl -s panTro2 -m 4 -d refseq -o
```

To create an index for Zebrafish danRer6 with refSeq for annotating indels

```
./main.pl -s danRer6 -m 5 -d refseq -o
```

Once the index has been created for a given species/assembly/annotation mode, Segtor can be run.

2.2.2 Creating custom databases using existing psl or bed files

To create a custom database with one or many psl or bed files describing the features, the following command :

```
./main.pl -s [species] -m [mode] -d [name of dabase] -o -e [comma-separated
    list of psl/bed files]
```

Please note that the only modes that are available for this are 1 and 2.
For instance, to build a custom database called “ncrna” using the files “ncrna1.psl” and “ncrna2.psl” on hg19, run:

```
./main.pl -s hg19 -m 1 -d ncrna -o -e ncrna1.psl,ncrna2.psl
```

The “ncrna” will be available for mode 1 and 2.

2.3 Running from the command line

To run Segtor, make sure main.pl is executable and run:

```
usage:
./main.pl -s [species code] -m [mode] -f [file] -d [database]
```

Here are the options:

	option	parameter	effect
Mandatory parameters			
	Species code		
	-s	[species code]	Code for species used by UCSC (e.g. hg19)
	Annotation mode		
	-m	[mode]	Annotation mode to use (e.g. -m 1) 1 Coordinate mode 2 Interval mode 3 SNVs 4 Find closest TSS 5 InDels/Translocations
	Input file		
	Specify either:		
	-f	[file]	Path to the file to annotate
	or		
	-l	[file of list of files]	File the full path of the files to annotate
	Gene database to use		
	-d	[database to use]	The name of the database to use ex: -d ests ESTs (mode 1,2 and 4 only) -d knowngene UCSC Known Genes (mode 1,2 and 4 only) -d refseq The annotation will be based on refSeqs -d ensembl The annotation will be based on Ensembl Genes
Parameters with default values			
	-c	[directory]	Use this directory to store indices and download the chromosomes/database files (default: $HOME/.segtor/)
	-r	[range]	Range(s) (for -m 1,2,3 only) (default: 0) in base pairs to use to annotate coordinates -r ”1000,5000”
Optional parameters			
	Annotation options		
	-b	none	Check dbSNP for existing snps (for -m 3 only, requires a large RAM)
	-a	none	Produce amino acid sequence (for -m 3 only, triggered by default for -m 5)
	-i	none	Do not check the reference base pair
	-n	none	Report inputs in terms of genes (genes.out file)
	-g	bins:size	for mode 4, with ”n:m”, report n bins of m base pairs each (ex: -g 10:1000 will create 10 bins of 1kb each side of a tss)
	-x	none	do not produce XML output files, used to save space
	-t	none	produce only a part of the total output (all.out), triggers -x
	-z	none	only produce stats
	Index building options		
	-o	none	Build the index files for the given species/database/mode (no annotation)
	-e	[files]	When using -o, specify a comma-separated list of files in psl or bed (with strand) format (from the UCSC Table Browser for example) to create the segment tree
	-b	none	Download dbSNP and build index for it (for -m 3 only)
	-y	none	Do not prompt prior to downloading the chromosomes/database files
	-p	[proxy server]	FTP proxy server to use (ex: http://127.0.0.1:21)

2.3.1 Examples of command line

To annotate coordinates for Human hg19 with ests at range 0

```
./main.pl −s hg19 −m 1 −d ests −f /path/to/input/file
```

To annotate intervals for Mouse mm9 with knowngene using a list of files at range 0

```
./main.pl −s mm9 −m 2 −d knowngene −l /path/to/fileOf/files
```

To annotate SNVs for Rat rn4 with ensembl for SNVs at ranges 1kb and 5kb and produce the mutated amino acid sequence:

```
./main.pl −s rn4 −m 3 −d ensembl −f /path/to/input/file −r ”1000,5000” −a
```

To annotate SNVs for Human hg18 with ensembl for SNVs and check dbSNP for existing SNPs and use /foo/bar to retrieve the indices

```
./main.pl −s hg18 −m 3 −d ensembl −f /path/to/input/file −b −c /foo/bar
```

To detect the closest TSS for Chimp panTro2 with refSeq and bin results into 5 bins of 10kb each

```
./main.pl −s panTro2 −m 4 −d refseq −f /path/to/input/file −g ”5:10000”
```

To annotate indels for Zebrafish danRer6 with refSeq and suppress the gene.out file

```
./main.pl −s danRer6 −m 5 −d refseq −f /path/to/input/file −n
```

2.3.2 Input format line

Mode 1, coordinate annotation:

```
   chromosome      coordinate        id
```

Example:

```
chr1      12393    coord1
```

Mode 2, interval annotation:

```
   chromosome      coordinate1       coordinate2       id
```

Example:

```
chr9      230894   231123   interval1
```

Mode 3, SNV annotation:

```
   chromosome      coordinate1       bpRef    bpRead   id
```

Example:

```
chrX      23094    A         C         snv1
```

Mode 4, closest TSS:

```
   chromosome      coordinate        id
```

Example:

```
chr19     903423   test1
```

Mode 5, indels/translocations:

```
INS chromosome          coordinate        DNAsequence       id
DEL    chromosome  coordinate1   coordinate2      id
TRANS   chromosome1      coordinate1       strand1  chromosome2        coordinate2
    strand2  id
```

Example:

```
INS  chr7        902344   CAGT     ins1
DEL     chr10  230984      230996   del1
TRANS   chr9     324023   +          chr4      23135     −          trans1
```

A script to convert from BAM, paired-end BAM, BED, snvmix and varscan to Segtor's native format can be found in the Segtor/scripts/ directory.