Name

chromoprocessor

Description

Processes multiple BAM files by splitting each BAM file into its regions. Parallelizes a command similar to

```
samtools mpileup 1.bam 2.bam [...] n.bam | java -jar VarScan > out.vcf
```

Dependencies

Python Modules

- **pysam**
- **argparse**
  * only if using a Python version < 2.7
- **futures**
  * only if using a Python version < 3.0

Other

- **vcftools**

Synopsis

usage:

```
chromoprocessor file_names action location [-h] [--n-region] [--verbose, -v]
```

Arguments

- `file_names`: a file containing the sames of the BAM files to process.
- `action`: the action for VarScan to run.
- `location`: the location of the VarScan jar.

Options

- `--n-region`: the number of regions to process in parallel. The default is two.
- `--verbose`: output additional information.

Examples

The easiset way to create the file with the name of the BAM files to process is with `ls` if you're going to call `chromoprocessor` from the same directory where the BAM files are, or with `find` otherwise, i.e `ls *.bam > to_process.txt`. Of course, the name of the file is completely arbitrary and can be anything.

After the file is produced it's as simple as

```
chromoprocessor to_process.txt mpileup2snp /home/You/VarScan.jar -v
```

If you have the hardware and you would like the BAM files to be processed quicker, you can run more jobs in parallel

```
chromoprocessor to_process.txt mpileup2snp /home/You/VarScan.jar -v --n-region=6
```

Notes

- A `varscan.conf` and a `samtools.conf` are expected to be in the current working directory when the program is called. The files should contain the arguments and parameters you want to pass to VarScan and samtools respectively. Examples of each can be find in the repository.

TODO

- add support for `--vcf-sample-list`
- add ability to specify which regions to process