SURPI

v1.0.0
March 2014

Note: For the most up to date version of the SURPI source code, go to this website: http://chiulab.ucsf.edu/surpi.

SURPI has been tested on Ubuntu 12.04. It will likely function properly on other Linux distributions, but this has not been tested.

Hardware Requirements

SURPI requires a machine with high RAM in order to run efficiently. This is mainly due to SNAP, which gains its speed by loading the reference databases completely into RAM. We've run SURPI successfully on machines with 60.5GB RAM. SURPI will use all cores on a machine by default, though the number of cores used can be adjusted within the config file. Much of SURPI is parallelized, so it benefits from using as many cores as possible.

The steps to install SURPI on a machine are as follows:

1. Install all software dependencies
2. Decompress SURPI.tar.gz, and place all files into a directory included in your $PATH
3. Create the databases
4. Customize certain SURPI files as shown below
5. Run SURPI

1. Install Software Dependencies

The below software must be installed for SURPI to function properly.

- fastQValidator
  http://genome.sph.umich.edu/wiki/FastQValidator
- Minimo (v1.6)
  http://sourceforge.net/projects/amos/files/amos/3.1.0/
- Abyss (v1.3.5)
  http://www.bcgsc.ca/platform/bioinfo/software/abyss
- RAPSearch (v2.12)
  http://omics.informatics.indiana.edu/mg/RAPSearch2/
- seqtk (v 1.0r31)
  https://github.com/lh3/seqtk
- SNAP (v0.15)
  http://snap.cs.berkeley.edu
- gt (v1.5.1)
  http://genometools.org/index.html

- fastq
  https://github.com/brentp/bio-playground/tree/master/reads-utils
- fqextract
  https://gist.github.com/drio/1168330
- cutadapt (v1.2.1)
  https://code.google.com/p/cutadapt/
- prinseq-lite.pl
  http://prinseq.sourceforge.net
- dropcache
  http://stackoverflow.com/questions/13646925/allowing-a-non-root-user-to-drop-cache

2. Decompress SURPI

Use tar to decompress SURPI.tar.gz. Something like the following should work:

```
tar xvfz SURPI.tar.gz
```

3. Create Databases

The content and creation of the SNAP databases is documented in the paper in the Reference Databases section, which is duplicated below:

> *A 3.1 gigabase (Gb) human nucleotide database (human DB) was constructed from a combination of human genomic DNA (GRCh37 / hg19), rRNA (RefSeq), mRNA (RefSeq), and mitochondrial RNA (RefSeq) sequences in NCBI as of March of 2012. The bacterial nucleotide, viral nucleotide, and viral protein databases used by SURPI in fast mode (bacterial DB, viral nucleotide DB, and viral protein DB, respectively) were also constructed from sequences in NCBI as of March of 2012. The 3 Gb bacterial DB was constructed from all bacterial RefSeq entries and consisted of 348,922 unique accessioned sequences, each with a minimum length of 100 bp. The 1.4 Gb viral nucleotide DB included 1,193,607 entries and was constructed by searching for all viral sequences in the 42 Gb National Center for Biotechnology Information (NCBI) nt collection using the query term "viridae[Organism]" in BioPython. The viral protein DB was similarly constructed by extracting viral sequences from the NCBI nr DB collection. Index tables for SNAP (v0.15) were generated with an empirically determined default seed size of 20 for the human DB and viral nucleotide DB, and seed size of 16 for the bacterial DB. Index tables for RAPSearch (v2.09) were generated from the viral protein DB using default parameters.*
>
> *To generate the National Center for Biotechnology Information (NCBI)*

nucleotide (nt) collection (NCBI nt DB) used by SURPI in comprehensive mode, the complete 42 Gb nucleotide collection (nt) was downloaded from NCBI in January of 2013. This collection consists of a comprehensive archive of sequences from multiple sources, including GenBank, European Molecular Biology Laboratory (EMBL), DNA Data Bank of Japan (DDBJ), and Protein Data Bank (PDB), and is the richest collection of annotated microbial sequence data publicly available. As SNAP uses 32-bit offsets in the reference genome during hashing, the aligner restricts the size of the reference genome to an absolute maximum of 2^32 bases, or ~4.2 Gb. Thus, the 42 Gb NCBI nt collection was first split into 29 sub-databases, each approximately 1.5 Gb in size. Each sub-database was then indexed separately by SNAP at default parameters with a seed size of 20. This generated 29 SNAP indexed databases, each approximately 27 GB in size, with the aggregate of all 29 databases referred to as the NCBI nt DB.

SNAP Databases

- Human DB

Comprehensive Mode

- NCBI nr DB
- Viral protein DB
- NCBI nt DB

Fast Mode

- Viral nt DB
- Bacterial DB

Taxonomy Databases

These databases can be created using the shell script titled: *create_taxonomy_db.sh*. Depending on your internet connection speed, and the speed of your system, this script may take several hours to complete creation of the database.

- gi_taxid_prot.db
- gi_taxid_nucl.db
- names_nodes_scientific.db

To use this script, execute the create_taxonomy_db.sh program. This script will download the necessary data from NCBI, and generate the above 3 databases.

4. SURPI file customization

Below are some notes on files that may need to be modified to run SURPI.

• **cutadapt_quality.csh** - specify location of /tmp folder

cutadapt_quality.csh defaults to using /tmp for temporary file storage. If using a system with limited space in this location, change the location to a directory with more storage space available.

• **taxonomy_lookup_embedded.pl**

Set database_directory to the location of the taxonomy databases created below.

• **tweet.pl**

SURPI has the ability to send out notifications via Twitter at various stages within the pipeline. If this feature is desired, you will need to set up a Twitter application within your account for this purpose. See https://dev.twitter.com/apps for more details.

Once an application has been set up, fill in the below parameters to the **tweet.pl** program.

```
consumer_key
consumer_secret
oauth_token
oauth_token_secret
```

• perl modules to install

```
Net::Twitter::Lite::WithAPIv1_1
Net::OAuth
```

5. Run SURPI

To run SURPI, execute the following in a directory containing your FASTQ input file.

1. This command will create the necessary config file to run SURPI:

```
SURPI_v22_15.sh -z <INPUTFILE>
```

After typing the above line, a config file and a “go” file will be created. The config file will contain default values for many parameters - these parameters may need to be modified depending on your environment. The config file has descriptions of the options allowed by SURPI.

2. Once the config file has been customized, the SURPI pipeline can be initiated by typing in the name of the go file that was created. Below is an example (boldfaced text is inputted by the user):

```
sfederman@tribble:/data/inputfile/test$ ls -laF
total 750212
drwxrwxr-x  2 sfederman sfederman      4096 Jan 20 16:45 ./
drwxrwxr-x 11 sfederman sfederman     61440 Jan 20 16:45 ../
-rw-rw-r--  1 sfederman sfederman 768143660 Jan 20 16:45 inputfile.fastq
sfederman@tribble:/data/inputfile/test$ SURPI_v22_15.sh -z inputfile.fastq

inputfile.config generated. Please edit it to contain the proper parameters for your
analysis.
go_ inputfile generated. Initiate the pipeline by running this program. (./
go_inputfile)

sfederman@tribble:/data/inputfile/test$ ls -laF
total 750220
drwxrwxr-x  2 sfederman sfederman      4096 Jan 20 16:47 ./
drwxrwxr-x 11 sfederman sfederman     61440 Jan 20 16:45 ../
-rw-rw-r--  1 sfederman sfederman      1976 Jan 20 16:47 inputfile.config
-rw-rw-r--  1 sfederman sfederman 768143660 Jan 20 16:45 inputfile.fastq
-rwxrwxr-x  1 sfederman sfederman        84 Jan 20 16:47 go_inputfile*
sfederman@tribble:/data/inputfile/test$ ./go_inputfile &
```

Progression of the pipeline can be followed by monitoring the log file (titled inputfile.SURPI.log, in the above example). We have also find it useful to monitor the status of the pipeline with the program *htop*.