Endovir

Report NCBI Visiting Bioinf

Jan P. Buchmann

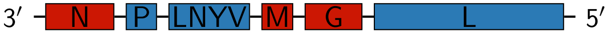
NCBI, NLM, NIH, Bethesda, MD, USA
The University of Sydney, Sydney, Australia

2017-12-08

Exogenous / Endogenous viruses

Exogenous virus

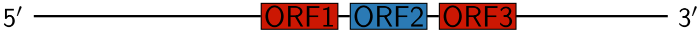
- Integration into host genome not required for replication
- Can occur occasionally, e.g. Human Hepres virus 6
- Clearly separated form host genome



3′ N P LNYV M G L 5′

Endogenous virus

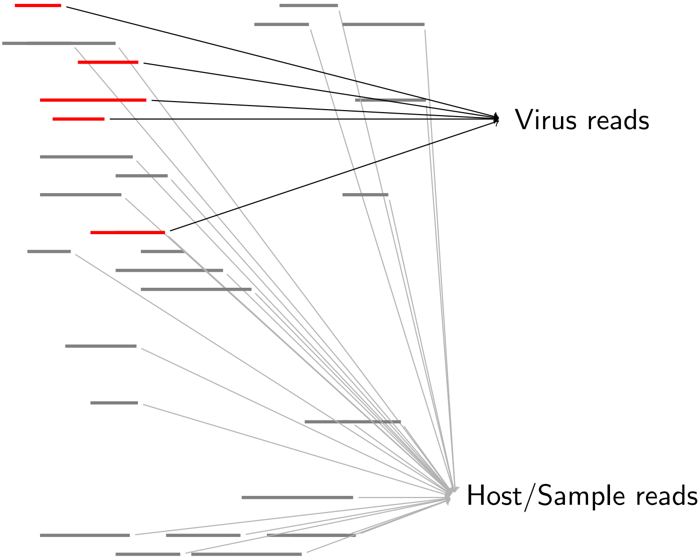
- Integration into host genome mandatory for replication
- Retroviridae, e.g. HIV
- Part of the host genome



5′ ORF1 ORF2 ORF3 3′

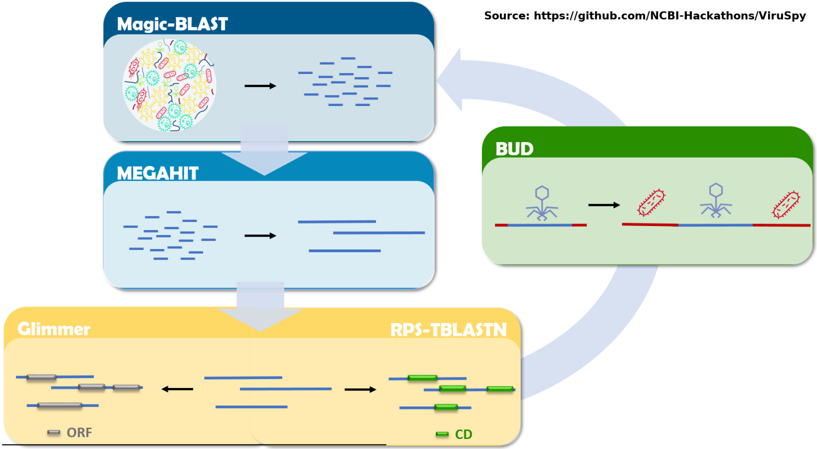
SRA data

- SRA data contains host and virus sequences
- Endogenous and putative novel viruses



Previous work

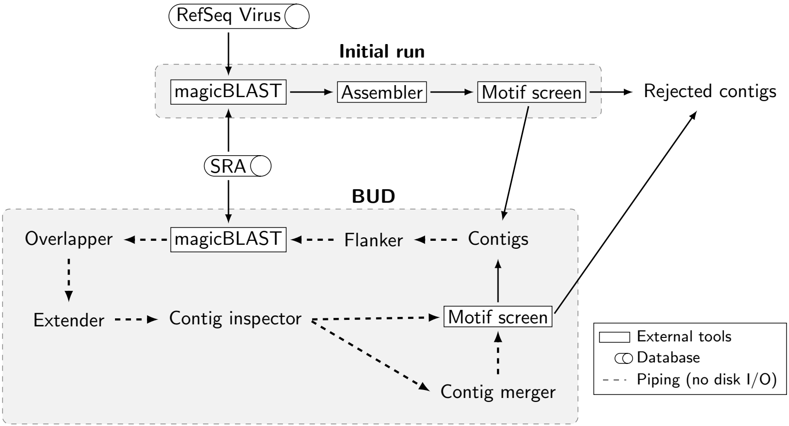
- ViruSpy [1]
- Logic mostly implemented in Bash and Perl
  → Complex to expand and adjust
- BUD algorithm



Source: https://github.com/NCBI-Hackathons/ViruSpy

[1] https://github.com/NCBI-Hackathons/ViruSpy

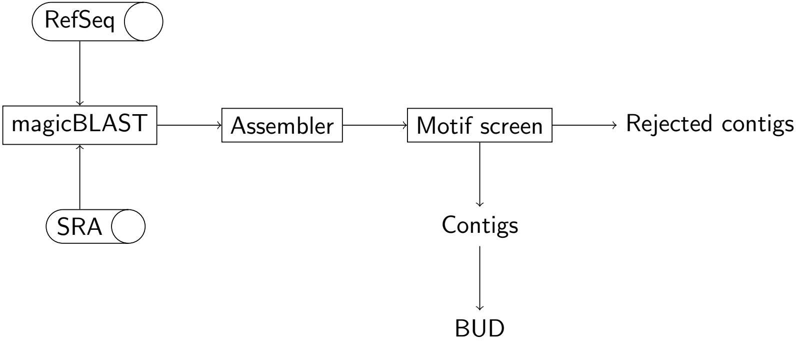
Endovir


RefSeq Virus
Initial run
magicBLAST
Assembler
Motif screen
Rejected contigs
SRA
BUD
Overlapper
magicBLAST
Flanker
Contigs
Extender
Contig inspector
Motif screen
Contig merger
External tools
Database
Piping (no disk I/O)

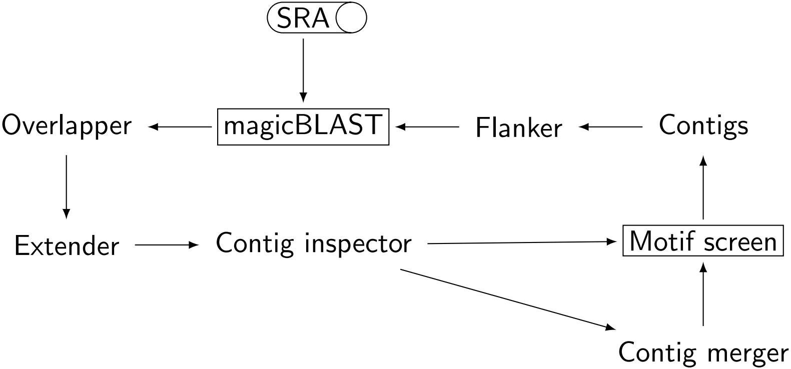
Usage

```
usage: endovir.py [-h] [-srr SRR] [--wd WD] [--max_cpu MAX_CPU]

Endovir

optional arguments:
  -h, --help            show this help message and exit
  -srr SRR              SRR number, e.g. SRR5150787
  --wd WD               Working directory for analysis
  --max_cpu MAX_CPU, -p MAX_CPU
                        Max number of cores to use. NOT YET IMPLEMENTED
```

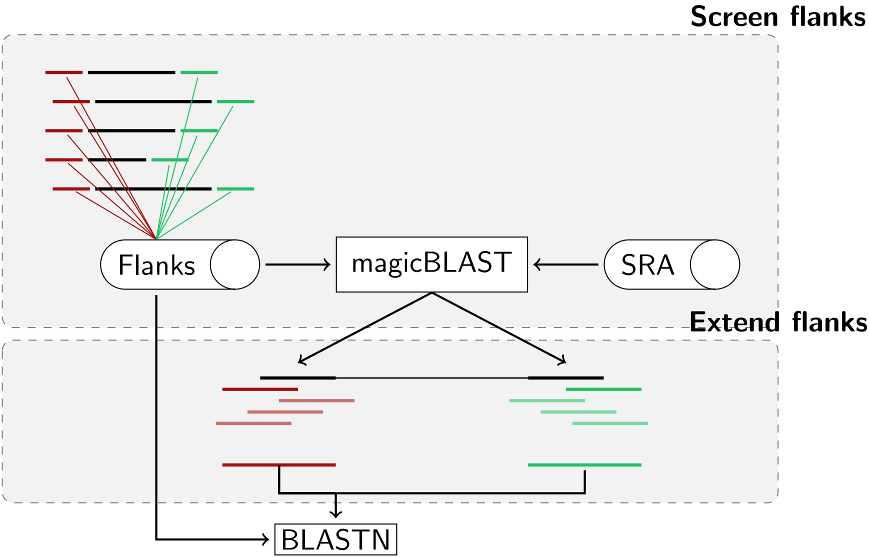
Init


RefSeq
magicBLAST
SRA
Assembler
Motif screen
Rejected contigs
Contigs
BUD

Bud


SRA
Overlapper
magicBLAST
Flanker
Contigs
Extender
Contig inspector
Motif screen
Contig merger

Flanker



Modularity

- Facilitating extending and adding other tools using classes which act as messenger between tool and Endovir.

```
class MagicblastAlignment(mapping_alignment.MappingAlignment):

  def __init__(self, cols):
    super().__init__(cols)
```

Modularity

```
class MappingAlignment:

  class Query:

    def __init__(self, name, start, stop, strand, qlen):
      self.name = name
      self.length = int(qlen)
      self.sra_rowid = name.split(’.’)[1]
      self.start = int(start) - 1
      self.stop = int(stop) - 1
      self.strand = 1 if strand == ’minus’ else 0
      self.aln_length = abs(self.stop - self.start) + 1

    def get_ordered_coords(self):
      if self.strand == 0:
        return (self.start, self.stop)
      return (self.stop, self.start)

  class Reference:

    def __init__(self, name, start, stop, strand):
      self.name = name
      self.start = int(start) - 1
      self.stop = int(stop) - 1
      self.strand = 1 if strand == ’minus’ else 0
      self.aln_length = abs(self.stop - self.start) + 1

    def get_ordered_coords(self):
      if self.strand == 0:
        return (self.start, self.stop)
      return (self.stop, self.start)

  def __init__(self, cols):
    self.qry = self.Query(cols[0], cols[6], cols[7], cols[13], cols[15])
    self.ref = self.Reference(cols[1], cols[8], cols[9], cols[14])
```

Outlook

- Adjust tool parameters on the fly
- Replace SRA toolkit with NCBI's ngs-python
- Test on more diverse datasets
- Docker