The B-55’s

# Abstract

*Human adenovirus B-55* (HAdV-B55) is a re-emergent pathogen that threatens dense populations.

# Introduction

The *Adenoviridae* family includes genera that infect a wide range of hosts and cell types. The *Mastadenovirus* genus includes species that infect mammalian hosts. They are globally distributed and are responsible for sporadic outbreaks in densely populated regions and close-living quarters. Symptoms range from acute respiratory disease to organ failure, depending on the viral species and host immune strength. Accordingly, individuals with developing or weakened immune systems account for most outbreak deaths.

# Methods

## Data

A Bash script coordinated the retrieval of nucleotide sequences and metadata in order to generate a local BLAST database. The Entrez Direct E-utilities scripts provided a command-line interface to query and retrieve data from the federated set of National Center for Biotechnology Information (NCBI) databases. The script generated the following query to download the sequences from the Nucleotide database.

```sql

(txid714978[PORG] OR txid343463[PORG]) AND biomol\_genomic[PROP] NOT gbdiv\_pat[PROP] NOT gbdiv\_syn[PROP]

```

Each square bracket occurs immediately after a search value and encloses a keyword that specifies the search field. The PORG and PROP fields correspond to primary organism and property respectively. The first two PORG terms use NCBI Taxonomy identifiers to limit the search to HAdV-55 and HAdV-11a respectively. The first PROP term sets the molecule type while the next two exclude GenBank divisions (Romiti and Cooper, 2011). Specifically, these terms limit the search to genomic DNA while excluding patents and synthetic constructs respectively. The \*\*esearch\*\* and \*\*efetch\* programs executed the query and retrieved the FASTA-formatted sequence data respectively. The script piped the results directly into the \*\*makeblastdb\*\* program to generate a BLAST database.

## Metadata

The same Bash script also coordinated the retrieval and normalization of sequence metadata to create a database of collection dates for subsequent Bayesian analysis. The script retrieved the set of accessions via \*\*blastdbcmd\*\* and piped them into the \*\*esummary\*\* to download the JSON-formatted metadata. The \*\*jq\*\* program processed the records from standard input using the following query.

```jq

.result | del(.uids) | map([.accessionversion, .subtype, .subname] | @tsv) | .[]

```

The query ran a series of filters to transform the JSON into a tab-separated file. The \*subtype\* and \*subname\* properties are pipe-delimited strings that correspond to keys and values respectively. An R script processed the data from standard input, using tidyverse functions to split the key-value pairs into new columns (Wickham, 2017). Another R script relied on the lubridate package to automatically convert the collection\_date field values into a consistent ISO-8601 format (Spinu et al., 2018). This series of commands generated a sorted tab-separated file mapping accessions to collection dates.

## Sequence Extraction

The full genome extraction method involved a series of piped commands. The \*\*blastdbcmd\*\* program dumped a space-separated list of accession-length pairs. Next, \*\*awk\*\* selected sequences with lengths greater than or equal to 34 kbp.

The gene extraction method required a query sequence. The \*\*blastdbmcd\*\* program generated the query for each gene based on a reference accession and sequence coordinates. A two-step alignment process extracted the set of homologous genes.

# References