

# PoMoCoV manual

Graham Jones, [www.indriid.com](http://www.indriid.com)

2020-05-16, June 25, 2020

## 1 Introduction

The package PoMoCoV and this manual are at a preliminary stage.

PoMoCoV is a BEAST2 package designed for phylodynamic analysis of SARS-CoV-2. It uses the polymorphism-aware model known as PoMo, which is also used by the BEAST2 packages PoMo [De Maio et al, 2015] and BADTRIP [De Maio et al, 2018]. PoMoCoV is a modified version of BADTRIP.

PoMoCoV is useful if you have sequencing data which captures within-host polymorphisms. It requires sequences of counts of the number of As, Cs, Gs, and Ts observed at each site, such as

..., 79-0-0-0,78-0-0-0,0-0-77-1,0-0-79-0, ...

A consensus sequence here would be ...AAGG... but PoMoCoV can make use of the fact the the third site is polymorphic. The PoMo model approximates the population with the Moran model for drift and mutation acting on a small virtual population.

### 1.1 Modifications for SARS-CoV-2

**Tree prior.** Optionally, a prior for the transmission tree can be added. The prior is based on observed transmission times for SARS-CoV-2.

**Mutation-only model within hosts.** Like BADTRIP, a drift-only model is used for the bottlenecks between hosts. In PoMoCoV, it is also possible to use a mutation-only model within hosts. Because the mutation rate is small and the time within hosts is short for SARS-CoV-2, the mutation-only model has a fast implementation.

**Unobserved hosts.** It is expected that SARS-CoV-2 data sets will have missing hosts. PoMoCoV uses a model which allows chains of unobserved hosts between the observed ones.

**New mutation models.** TODO

**Drift-only speedup.** This is a speedup of the implementation which does not change the model. (It is mainly useful when the mutation-only model is used, otherwise the time is dominated by the within-host calculations.)

### 1.2 Installation and requirements

PoMoCoV requires BEAST 2.6 (which requires Java 8 or later) and should normally be installed using BEAUTI.

Since PoMoCoV is not yet peer-reviewed, it does not appear in Beauti by default. You need to add the following repository: <https://raw.githubusercontent.com/CompEvol/CBAN/master/packages-extra.xml>.

Choose **File**→**Manage packages**, then **Package repositories** and paste in the above URL. Then you should be able to install, upgrade or uninstall PoMoCoV in the usual way.

### 1.3 Running PoMoCoV

You can run PoMoCoV using the BEAST2 GUI, but (at least on Windows) it is very slow to get started, and much faster using the command line. See <https://www.beast2.org/2019/09/26/command-line-tricks.html>. In brief, you can install BEAST 2.6, and use a command like:

```
java -jar C:\Users\GRJ\AAA\software\BEAST262\lib\launcher.jar -working -overwrite PoMoCoV-std.xml
```

`-working` tells BEAST2 to put the output files (log and tree files) in the same directory as the XML (I recommend this). `-overwrite` tells BEAST2 to overwrite any previous output files. If you want to resume an analysis, replace `-overwrite` with `-resume`.

## 2 Creating an XML file for PoMoCoV

PoMoCoV does not work with Beauti, so you have to create an XML file, in a similar way to BADTRIP. This section explains the differences from BADTRIP and the new model implemented in PoMoCoV.

---

```
<alignmentPoMo spec="beast.evolution.alignment.AlignmentPoMo" id="alignment" dataTypePoMo="PoModata">

    <acgtCounts spec="SequencePoMo" id="s1001" taxon="t1001" value="50-0-0-0, 49-1-0-0, [...]
```

Same as BADTRIP, except that `sequence` is changed to `acgtCounts`. (A change was necessary to make PoMoCoV compatible with BEAST 2.6.)

---

```
<TraitSetPoMo spec='pomocov.TraitSetPoMo' id='traitSet'
    direction="forward" units="day" unobservedHosts="true"
    samplesHostsValue="t1001=H1, t4001=H4, t5001=H5, t2001=H2, t7001=H7"
    samplesDatesValue="t1001=9.9, t4001=12.2, t5001=9.4, t2001=11.2, t7001=7.4"
    HostDatesStartValue="H1=-0.1, H4=2.2, H5=-0.6, H2=1.2, H7=-2.6"
    HostDatesEndValue="H1=10.4, H4=12.7, H5=9.9, H2=11.7, H7=7.9">
    <taxa spec='beast.evolution.alignment.TaxonSet' alignment='@alignment'/>
</TraitSetPoMo>
```

Same as BADTRIP, except that `unobservedHosts="true"` can optionally be added. This makes the model more flexible by allowing chains of unobserved hosts between the observed hosts. It also adds an extra unobserved primary case for the outbreak. If `unobservedHosts` is true, all the observed hosts must have at least one sample, and (finite) start and end dates.

---

```
<siteModel spec="pomocov.SiteModelPoMo" id="siteModel">
    <mutationRate spec='RealParameter' id="mutationRate" value="0.000003" lower="0.0"/>
    <substModel spec="pomocov.PoMoGeneral" virtualPop="3" estimateRootFreqs="false"
        useTheta="False" theta="0.0" id="PoMo.substModel">
        <fitness spec='RealParameter' id="PoMo.fitness" value="1.0 1.0 1.0 1.0" lower="0.8" upper="1.2"/>
        <rootNucFreqs spec='RealParameter' id="PoMo.rootFreqs" value="0.25 0.25 0.25 0.25"/>
        <mutModel spec='pomocov.MutationModel' id="PoMo.mutModel" modelType="HKY" mutationOnlyModel="true">
            <rateVector spec='RealParameter' id="PoMo.mutRates" value="0.33 0.33"/>
            <nucFreqs spec='RealParameter' id="PoMo.nucFreqs" value="0.25 0.25 0.25 0.25"/>
        </mutModel>
    </substModel>
</siteModel>
```

Similar to BADTRIP, except that `modelType` must be supplied and, optionally, `mutationOnlyModel` can be used. (BADTRIP used the length of `rateVector` to infer the mutation model, but PoMoCoV may have two models with the same number of parameters.)

If `mutationOnlyModel` is true, the `mutationRate` will be smaller, and `PoMo.mutRates` will be larger, so a different starting values are appropriate. Note that larger values for `PoMo.mutRates` can result in problems for the old model.

---

```
<input spec='CompoundDistribution' id='parameterPriors'>
    <distribution spec='beast.math.distributions.Prior' x="@PoMo.mutRates">
        <distr spec='LogNormalDistributionModel' M="-1.0" S="1.0"/>
    </distribution>
```

```

<distribution spec='beast.math.distributions.Prior' x="@mutationRate">
  <distr spec='LogNormalDistributionModel' M="-13.0" S="2.0"/>
</distribution>
<distribution spec='beast.math.distributions.Prior' x="@PoMo.bottleneck">
  <distr spec='Exponential' mean="4.0" offset="1.01"/>
</distribution>
</input>

```

Same as BADTRIP, except that I have changed the priors, since the meaning of the parameters has changed. In particular, note that the bottleneck must be at least 1.0 (and cannot realistically be extremely close to 1).

---

```

<input spec='pomocov.TreePoMoCoVPrior' id="treePrior" tree="@tree" unobservedCountPriorMean="1.0" >
</input>

```

New, and optional. Provides a prior for the transmission tree based on observed distributions for transmission times [Ferretti et al, 2020]. This is a built-in prior with no hyperparameters (at the moment). It uses Burr distributions with various hyperparameters to approximate “waiting times” [Ogbunugafor et al, 2020] between hosts, transmission times after infection within one host, and durations of chains of unobserved hosts.

---

```

<input spec='pomocov.PoMoCoVTreeLikelihood' id="treeLikelihood" treePoMo="@tree">
  <seqError spec='RealParameter' id="PoMo.seqError" value="0.0" upper="0.01"/>
  <bottleneck spec='RealParameter' id="PoMo.bottleneck" value="3.0" lower="1.01" upper="1000"/>
  <data idref="alignment"/>
  <siteModel idref='siteModel'/>
</input>

```

PoMoCoVTreeLikelihood replaces TreeLikelihoodPoMoTransmission or TreeLikelihoodPoMoTransmissionSlow used by BADTRIP. PoMo.bottleneck is constrained to be above 1.0.

---

```

<run spec="MCMC" id="mcmc" chainLength="1000000" storeEvery="50000">
  <state>
    [...]
  </state>
  <distribution spec='CompoundDistribution' id='posterior'>
  <distribution idref="treeLikelihood"/>
  <distribution idref="treePrior"/>
  <distribution idref="parameterPriors"/>
</distribution>

```

Same as BADTRIP, except that <distribution idref="treePrior"/> added.

---

```

<!-- Operators for continuous parameters-->
<operator spec='ScaleOperator' id='mutScalerAll' parameter="@PoMo.mutRates"
  scaleFactor="0.95" weight="5" scaleAll="true"/>
<operator spec='ScaleOperator' id='mutScalerOne' parameter="@PoMo.mutRates"
  scaleFactor="0.98" weight="5" scaleAll="false"/>
<operator spec='ScaleOperator' id='errScaler' parameter="@PoMo.seqError"
  scaleFactor="0.95" weight="5"/>
<operator spec='ScaleOperator' id='muRateScaler' parameter="@mutationRate"
  scaleFactor="0.95" weight="5"/>

```

```

<operator spec='ScaleOperator' id='bottleneckScaler' parameter="@PoMo.bottleneck"
                                scaleFactor="0.9" weight="5"/>
<operator spec='UpDownOperator' id='RelativeScalerMuHKY' scaleFactor="0.95"
                                weight="5" down="@mutationRate" up="@PoMo.mutRates"/>
<!-- Tree operators -->
<operator spec='pomocov.UniformTransmission' weight="50" tree="@tree"/>
<operator spec='pomocov.WilsonBaldingTransmission' weight="50" tree="@tree"/>
<operator spec='pomocov.WilsonBaldingNewHeight' weight="50" tree="@tree"/>
<operator spec='pomocov.ParentChildExchange' weight="50" tree="@tree"/>
<operator spec='pomocov.PoMoCoVCountsWalker' weight="50" tree="@tree"/>

```

I added the `scaleAll="true"` version of `ScaleOperator` for `PoMo.mutRates` (maybe helps mixing a bit). The `SubtreeSlideTransmission` operator is omitted (not yet implemented for `PoMoCoV`). The `PoMoCoVCountsWalker` operator is added for changing the number of unobserved hosts linking a host to the originating observed host.

The tree operators `PoMoCoVCountsWalker`, `UniformTransmission`, `WilsonBaldingNewHeight`, `WilsonBaldingTransmission`, and `ParentChildExchange` take note of the `unobservedHosts` flag in `TraitSetPoMo`. `PoMoCoVCountsWalker` does nothing and the other three revert to the `BADTRIP` versions if `unobservedHosts` is false. They have new implementations when `unobservedHosts` is true.

---

```

<logger logEvery="1000" fileName="transmission-uo.log">
  <model idref='posterior' />
  [...]
</logger>

<logger logEvery="1000" fileName="transmission-uo.trees" mode="tree">
  <log idref="tree" />
</logger>

<logger logEvery="1000">
  [...]
</logger>

```

I have just changed file names and screen output. The new parameters (originating times and counts of unobserved hosts) are in the trees file as metadata, like this:

```

[&host=H4,height=7.8687898608387945,originHeight=12.95367165114677,unobservedCount=1,
highestSampleTime=0.5,start=10.5,end=0.0]

```

The other values for `highestSampleTime`, `start`, and `end` are copied from the input XML. They are added for convenient analysis of the results.

---

## 3 Changes from previous versions

### Changes in v0.2.0

- Fixed serious bug in calculation for mutation-only model.
- The bottleneck parameter is now the actual bottleneck size, instead of the ‘bottleneck intensity’ used by `BADTRIP`.

- For the mutation-only model, the mutation rate is now the actual mutation rate in substitutions per site per day. The `PoMo.mutRates` parameter (a vector of relative rates such as transitions and transversions in HKY model) is normalised before usage. [I never fully understood what these rates meant in BADTRIP.]
- For the mutation-only model, the root frequencies are now estimated using the bottleneck and the mutation rate, together with the nucleotide frequencies supplied in the XML, and a built-in value for the typical transmission time (about 5 days).
- The calculation of these root frequencies now uses the beta-binomial distribution, not the Watterson estimator.

## v0.1.0

First release.

## References

- Nicola De Maio, Dominik Schrempf, Carolin Kosiol, *PoMo: An Allele Frequency-Based Approach for Species Tree Estimation*, Systematic Biology, Volume 64, Issue 6, November 2015, Pages 10181031, <https://doi.org/10.1093/sysbio/syv048>
- De Maio N, Worby CJ, Wilson DJ, Stoesser N (2018) *Bayesian reconstruction of transmission within outbreaks using genomic variants*. PLOS Computational Biology 14(4): e1006117. <https://doi.org/10.1371/journal.pcbi.1006117>
- Ferretti et al, *Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing*. <https://science.sciencemag.org/content/early/2020/04/09/science.abb6936>
- Ogbunugafor et al, *The intensity of COVID-19 outbreaks is modulated by SARS- CoV-2 free-living survival and environmental transmission*, <https://www.medrxiv.org/content/10.1101/2020.05.04.20090092v2>