Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Time invariant analysis of epidemics with EpiCompare

Shannon K. Gallagher
Biostatistics Research Branch
National Institute of Allergy
and Infectious Diseases

Benjamin LeRoy
Dept. of Statistics & Data Science
Carnegie Mellon University

Abstract

We present **EpiCompare**, an R package that suppliments and enhances current infectious disease analysis pipelines and encourages comparisons across models and epidemics. A major contribution of this work is the set of novel *time-invariate* tools for model and epidemic comparisons - including time-invariate prediction bands. **EpiCompare** embraces R's *tidy* coding style to make adoption of the package easier and analysis faster. This paper provides an overview of both the tools in and intuition behind **EpiCompare** and a thorough demonstrating of the tools through a detailed example of a full data analysis pipeline.

Keywords: keywords, not capitalized, Java.

1. Introduction

The recent (and on-going) COVID-19 global pandemic has galvanized public interest in understanding more about infectious disease modeling and has highlighted the usefulness of research in the area of infectious disease epidemiology. Infectious diseases inflict enormous burdens on the world: millions of lives lost and trillions of dollars spent yearly. Infectious disease models typically attempt to do one or more of the following: 1) predict the spread of current and future epidemics (e.g. flu prediction Biggerstaff *et al.* 2016), 2) analyze past and current epidemics to increase scientific knowledge (e.g. historical measle outbreaks Neal and Roberts 2004), and 3) forecast or project epidemic scenarios under pre-specified parameters (e.g. Ferguson *et al.* 2020). At the same time, descriptive statistics and visualizations from universities, many branches and levels of government, and news organizations are an important first step of the process as has been seen in the current COVID-19 epidemic(Dong *et al.*

2020; CDC 2021; The Washington Post 2021). [1]

With the many visualization and exploratory tools, models and modeling paradigms, and reviews and comparisons in the literature and through the MIDAS (Models of Infectious Disease Agent Study) network (MIDAS Network 2021), this field has a lot of devices to aid an individual practitioner decide the correct approach. For example, R packages such as **surveillance**, **EpiModel**, and **pomp** have all made significant steps in standardizing the flow of the data analysis pipeline for epidemic modeling through digitizing data sets, making accessible statistical models, and providing a plethora of educational material for both coding novices and experts alike (Meyer *et al.* 2017; Jenness *et al.* 2018; King *et al.* 2016).

At the same time, analysis packages often only address a specific portion of the analysis pipeline~~, for instance focusing on certain types of models.~~ These modeling tools~~, which~~ usually require learning package-specific syntax~~,~~ and often don't provide easy ways to compare and assess their models on new data. Moreover, exploring, ~~and~~ modeling and comparing epidemics require transforming and *tidying* data in different ways. To fill these gaps, we present our R package **EpiCompare**. Our package's primary focus is to aid and advance research in the area of comparison and assessment of epidemic and epidemiological models. In Figure 1, we illustrate the data analysis pipeline of infectious diseases as 1) data pre-processing, 2) exploratory data analysis (EDA), 3) modeling and simulating, 4) post-processing, and 5) comparison and assessment; where each previous part of the pipeline influences the next. **EpiCompare** provides tools to aids practitioners in all areas of this pipeline.

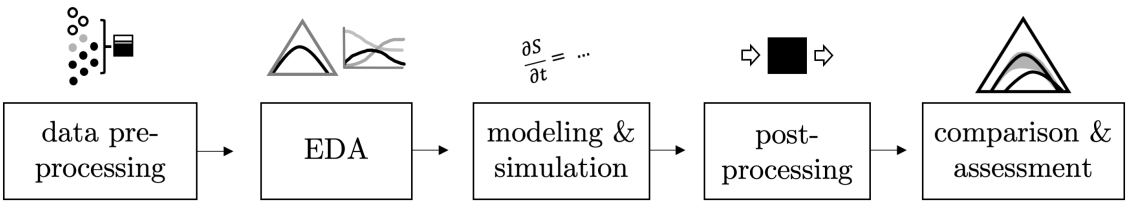

Figure 1: An idealized epidemiological data analysis pipeline.

[One of **EpiCompare**'s main contribution to] comparison and assessment of epidemics is through tools that provide *time-invariant* assessments. Epidemics, despite being defined a a process that evolves overt time, often needs to be compared in a way not constrained to initial times or time scales to understand the processes at play.] [With time-invariant analysis, comparing decades-long outbreaks of HIV in the US to a 10 day outbreak of norovirus on a cruise ship is possible.][2] Compared to time-dependent comparison tools for state-space modeling, [time-invariant analysis can make it easier to compare state-space ~~models~~ epidemic representations in a more global, holistic fashion.] Many time-dependent comparison tools only [examine the proportion of individuals in each state (at a given time) in a piece-wise / marginal fashion.] [These time-dependent approaches can reduce the amount of ~~connections~~

[1] [Ben says: probably should have a conclusion sentence here - seems to end abruptly. *This is less so the case now.]

[2] [Ben says: brought this sentence from a later paragraph as it's much more introductory.]

~~that can be seen~~ insights that can be drawn, similar to examining projections of a multidimensional distribution onto a single axis one at a time. Tools in **EpiCompare** give the user the ability to extend their toolkit to evaluate epidemics within a time-invariant lens. At the same time, the goal of **EpiCompare** is not to supplant existing infectious disease modeling tools and software but, rather, is a concerted effort to create standard and fair comparisons among models developed for disease outbreaks and outbreak data.]

This paper is broken up into the following sections; section **??** motivates and showcases tools of time-invariant analysis, section 3 presents an outline of how **EpiCompare** aids a practitioner in every step of the pipeline and section 4 provides a ~~thorough~~ demonstrating of the tools through a detailed example of a full data analysis pipeline.

2. Time-invariant analysis

[EpiCompare emphasizes the value of analyzing epidemics in a *time-invariant* way - approaches that remove some or all of the impact of start/end times and recording time scales when preforming the analysis. In this section we will highlight weaknesses of time-*de*pendent analysis and define the mathematical underpinning of the time-invariant approach we take. To accomplish this goal we will first demonstrate the inability of time-dependent tools to well quantify a classic epidemic parameter, the reproduction number R_0. Then we will motivate new time-invariant approaches for more complex situations where R_0 doesn't capture the complexities of outbreaks. This will lead us to our final subsection that defines ways to view epidemics in a time-invariant lense and discusses natural properties of these representations that allow for clearer ways to compare models and epidemics.

2.1. Motivation of time-invariant analysis through the reproduction number R_0

A lot of epidemiological research is interested in understanding intrinsic properties of the epidemic, not depending upon exactly when the epidemic occurred or how frequently data was collected. One of the most famous numerical summaries of an epidemic is the reproduction number - R_0, a time-invariant value that is [defined as the expected number of infections caused by a single infect or who is added to a completely susceptible population.] Interestingly, Gallagher *et al.* (2020) has shown that the estimation of R_0 [can be sensitive to time-dependent parameters] like estimation of [the beginning and end of an epidemic]. This time-dependent sensitivity in estimating R_0 reflects a general problem in epidemiology surrounding how to transform time-dependent information into insights into desirable properties of an epidemic.

In many situations, epidemiologists want a much deeper understand of an epidemic than just the reproducible number. One common time-dependent tool is a series of time series line plots that track the proportion of the population in a given state (e.g. infected) at a given time. Figure 2 visualizes two different simulated epidemics with population states (s)usceptible, (i)nfectious and (r)ecovered. These simulations are actually generated under a discrete approximation of Kermack and McKendrick (1927)'s SIR model. [This model captures the transitions from one state to the next as a system of ordinary differential equations, where N is the total number of individuals, β is the rate of infection, and γ is the rate of recovery,]

$$
\begin{aligned}
S'(t) &= -\frac{\beta S(t) I(t)}{N} \\
I'(t) &= \frac{\beta S(t) I(t)}{N} - \gamma I(t) \\
R'(t) &= \gamma I(t).
\end{aligned} \tag{1}
$$

[From this model, the reproduction number is the ratio of the infection rate to the recovery rate, $R_0 = \beta/\gamma$.]

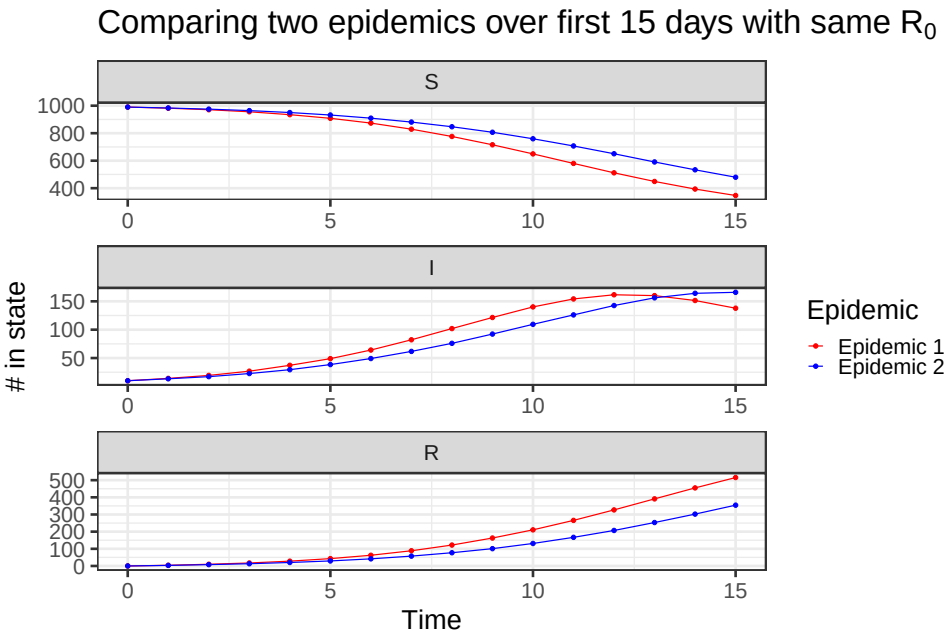

Figure 2: Example of two epidemics with different β and γ parameters but the same initial reproduction number $R_0 = 2$. Both epidemics are generated from models with $N = 1000$ individuals with $S(0) = 990$ and $I(0) = 10$.

Even though Figure 2's visual analysis might be able to present more complex properties of the epidemic, one might wish to understand how these two simulated epidemics' R_0 values compare. Even though these two simulated epidemics appear different in these figures ([including having different infection peaks]), there is no real intuitive way to compare the epidemic's R_0. In fact, both of these simulated epidemics started out the same population (1000 people with 10 infected) and have the same R_0 value, just with slightly different infection and recovery rates ($\beta_1, \gamma_1 = 0.8, 0.4$, and $\beta_2, \gamma_2 = .64, .32$). Even with simple generative models, this time-dependent visualization tools is not able to be used to compare a very simple but powerful numerical summary.

2.2. Ternary plots: a time-invariant visualization tool

The faceted time series plot, like that seen in Figure 2, not only fails to allow the practitioner to compare simulated epidemics' R_0 values but also presented epidemics' trajectory data so

that only the marginal information for each state could be examined at a given time. A more holistic approach to visualizing the overall path of our simulated epidemics would be to examine each epidemic traverses the three-dimensional $(S(t), I(t), R(t))$ space. Figure 3's left plot does just that, presenting the trajectories of the simulated epidemics in this three-dimensional space. [For state space models like in our example, given the constraint that $S(t) + I(t) + R(t)$ is always equal to N (the total population size) we can visual these point in a a two-dimesional *ternary* plot, as seen in Figure 3's right plot.] In this both of Figure 3's plots we an observe that the epidemics are on the same trajectory. In this simple generative example setting, [this indicates that the two epidemics have the same R_0 value]. This can be proven mathematically.

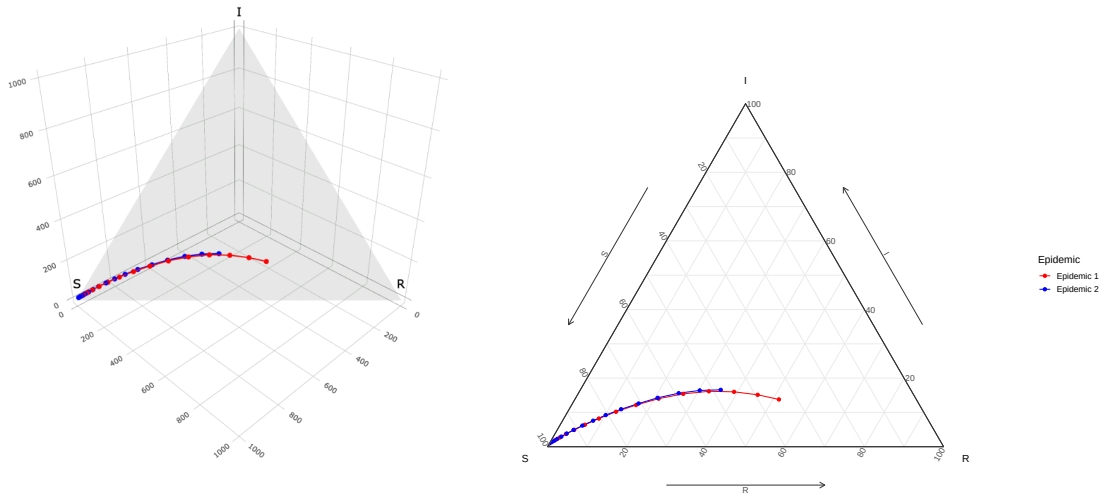

Figure 3: Left: trajectory of epidemic in three-dimensional space, plotting $(S(t), I(t), R(t))$. Right: the gray-shaded region and epidemic trajectories shown from (left) now shown in two-dimensional space. This is more commonly known as a ternary plot.

Theorem 1. *Let there be two Kermack and McKendrick (1927)'s SIR models* $((S_1(t), I_1(t), R_1(t))_{t\geq 0}$ *and* $(S_1(s), I_1(s), R_1(s))_{s\geq 0})$*, with* $(S_1(0), I_1(0), R_1(0)) = S_2(0), I_2(0), R_2(0))$ *Let both models have the same* R_0 *(aka* $R_0 = \frac{\beta_1}{\gamma_1} = \frac{\beta_2}{\gamma_2}$ *and define* $a > 0$ *such that* $\beta_2 = a\beta_1$*. [Then for all* $t > 0$ *there exists an* $s > 0$ *such that* $(S_1(t), I_1(t), R_1(t)) = (S_2(s), I_2(s), R_2(s))$*. Moreover,* $s = \frac{1}{a}t$*.]*

[The proof of Theorem 1 relies on a fairly recent result from Harko *et al.* (2014) and is shown in detail in Proof 4.1. The consequence of Theorem 1 is that for two SIR models that have the same initial percent of individuals in each state and R_0 then for every point on the epidemic path of the first SIR model can be mapped to a point on the epidemic path of the second SIR model. In other words, the two epidemics form the same filamental trajectory.]

2.3. Time-invariance beyond SIR models: Trajectories and Filaments

[Through the R_0 example, we see that treating epidemics like filamental trajectories embedded in a lower dimensional space allows us better compare the overall structure of the epidemic and

see how the population is directly impacted.] In this section we present time-invariant tools that can be applied to complex epidemics where the epidemic's generative process is unknown and can have more than three states. These tools leverage the idea that an epidemic can be viewed as a trajectory and that many properties of the epidemic are well captured when we do so. This approach is useful when the epidemic of interest has only gone [through a single realization of its outbreak (before the population of individuals become susceptible again)].

The first set of tools allows a practioner to define distances between epidemics whose time features don't well align. For completed epidemics, one way to better examine their properties is to represent their filamental trajectories as a finite sequence of equally spaced points. [This representation induces a natural distance ~~between this type of representation~~ between epidemics, specifically:]

$$d_{\text{equi-distance}}(\psi_1, \psi_2) = \int_{s\in[0,1]} (\tilde{\psi}_1(s) - \tilde{\psi}_2(s))^2 ds$$

where $\tilde{\psi}_i(s)$ the point along ψ_i that is $s \cdot |\psi_i|$ distance away from the start of ψ_i. This distance is naturally time-invariant, and can be plugged into multiple distance-based assessment tools to examine the overall "extremeness" of points, including pseudo-density estimators and depth/local depth functions (for examples see Ciollaro *et al.* 2016; Geenens and Nieto-Reyes 2017). These extremeness estimators can be useful when comparing the true epidemic to a a set of simulated epidemics,] and practitioners can interpret the univariate epidemic's extremeness score relative to the extremeness scores of the simulations very easily.

When the goal is to predict the future impact / trajectory of a epidemic, time-invariant tools can help describe state-space regions in which we expect the true epidemic to traverse. In settings where the epidemic only goes a single outbreak, these regions can be very telling if simulation models well capture the epidemic's structure. In **EpiCompare** we create geometric prediction regions around all but the α proportion of most extreme simulated epidemics. These geometric regions can also be used to compared simulation models that have different time scales, parameters and even different statistical philosophies, through set different distances. Although visualization is the easiest when the epidemic has three states, prediction regions can be useful to assess and compare simulation models where the epidemics have multiple states.

Overall, there are many tools to aid in the assessment and comparison of epidemics and models that avoid being affected by time-based parameters. We believe the time-invariant analysis provides many insights and should be in the toolkit of many epidemiologists. **EpiCompare** provides a strong starting point to do just that.

3. Overview of EpiCompare

In this section, we present the tools implemented in **EpiCompare** and explain how they aid in the data analysis pipeline. In Fig. 4, we illustrate how our package's functions fit into the data analysis pipeline introduced in Fig. 1. All front-facing functions are aimed to be as user-friendly as possible. We also focus on providing the user "tidyverse" style functions, that encourage piping objects from one function to the next and follow clear "verb" naming schemes (Wickham *et al.* 2019). Although users can incorporate **EpiCompare** into any step in the data analysis pipeline, there are two primary points of entry. The first point of entry is the

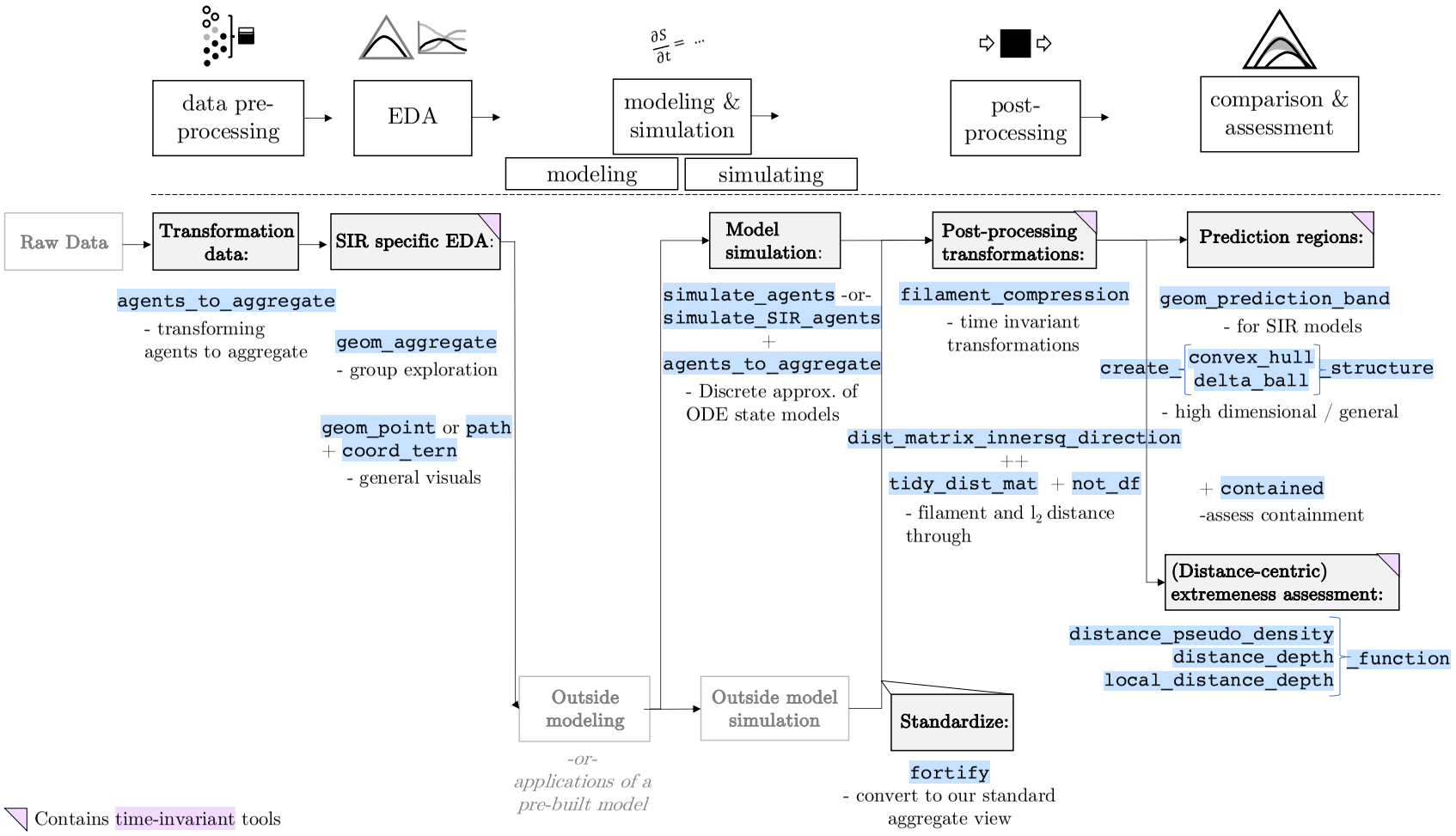

Figure 4: How **EpiCompare** supplements and aids in the epidemiological data analysis pipeline.

very beginning with pre-processing and visualizing raw data, and the second point of entry is after modeling and simulation. Figure 4 captures these different paths, and we highlight[3] both approaches and how to leverage **EpiCompare** in the subsections below.

Data Pre-processing

The first step of most data analysis is "cleaning" the data to a format that is friendly for both computers and programmers[4] so it can be explored. ~~There are multiple ways to collect epidemiological data.~~ In epidemology, there are multiple different formats the data can arrive in. [5] Sometimes individual records are collected, with times of different states of the epidemic (infection, recovery, etc.) as well as individual information like network structure, location, and sub-population information. Other data collections focus on aggregate counts of individuals in each epidemic state. In fact, many times only the number of new infections at each time step (e.g. weekly case counts) is observed. Compartment totals (amounts of individuals in each state) are then imputed from those case counts ~~along with~~and[6] other information about the disease and the population of interest. In **EpiCompare**, we focus on understanding the overall impact of an outbreak at the aggregate/population level, which allows for streamlined examination of overall trends of an epidemic.

To help the practitioner examine epidemics from an aggregate/ population lens, we provide a function called `agents_to_aggregate()`. This function transforms data about individual/agents' initial entry into each state (e.g. start of infection, start of recovery, etc.) to an aggregate view of how many individuals were in a state at a given time. There are often situations where ~~grouping~~ aggregating[7] agents into subpopulations (e.g. sub~~populations~~groups[8] defined by age or sex) can highlight different aggregate level trends. For example, research by Rvachev and Longini (1985); Anderson and May (1992); Worby *et al.* (2015) develop state-based models that account for differing disease dynamics in different subpopulations. In **EpiCompare**, we facilitate subpopulation analysis by combining the function **dplyr**::`group_by()` and `agent_to_aggregate()` to provide aggregation ~~at a~~by[9] group level.

The `agents_to_aggregate()` function is flexible and can deal with a wide range of information about each individual. ~~It can,~~ In fact, this function can account for infinitely many states. This functionality allows the practitioner to aggregate information relative to common states (e.g. "Susceptible", "Infectious", and "Recovered") as well as more complex states (e.g. "Exposed", "iMmune", "Hospitalized"). Additionally, `agents_to_aggregate()` permits indicators for death/exit and birth/entry dates. Overall, this function is a powerful tool for pre-processing data, and it lowers the barrier for entry into data analysis for less experienced practicioners.[10]

---

[3] [Ben says: we need to make sure we actually do highlight]

[4] [Ben says: I'm unsure this is needed and naturally adds more text. Willing to accept...]When in doubt, remove.

[5] [Shannon suggests: "Epidemiological data are collected in many different formats."] [Ben says: I see the point you're trying to make and made a new edit - but it's less connected to the previous sentence since "format" is the end.]What about adding a linking sentence "Before data can be explored, it must be colleected. Epidemiological data are collected in many different formats."

[6] [Ben says: Declined. I'm unclear why we should. Is the sentence unclear? Maybe it's the way it's a compound sentence? - If so a rewrite would be better?]Upon reflection, I think it's fine.

[7] [Ben says: reason - connect to languare from sentence before.]fine

[8] [Ben says: declined. I'm unclear why we'd do this.]fine

[9] [Ben says: declined. unclear about this change. Also would it not be "by groups"? Orignial seems to flow more.]fine

[10] [Shannon says: I added this in theme with 'highlighting a point of entry'] [Ben says: I'm unclear about

EDA

~~With raw data, "Getting to know" our~~ In the early stages of a project, getting to know the[11] data currently means figuring out useful combinations of visualizations ~~,~~and numerical summaries and ~~subsets~~groupings~~s~~exploring different groupings of the data[12]. ~~An expert coder has many ways to successfully explore the data in an aggregate lens using~~ `agents_to_aggregate()`~~.~~ ~~For less experienced coders,~~ **EpiCompare** ~~also includes tools to rapidly explore data that has three epidemic states.~~ An expert coder can start from `agents_to_aggregate()` to successfully accomplish EDA in many ways, but **EpiCompare** also includes tools that allow a novice coder to rapidly explore data, as long as there three unique epidemiological states (like the SIR model).[13] Building on ~~the tools in~~ **ggplot2** and **ggtern** packages, **EpiCompare**'s `geom_aggregate()` provides a rapid way to explore different subpopulations' experiences of an epidemic (Wickham 2016; Hamilton and Ferry 2018). The function `geom_aggregate()` provides a visualization tool to holistically examine aggregate level information across different subpopulations by visualizing each subpopulation's epidemic trajectory in the three-dimensional state space. [14] Visualization tools for three-state models were developed because (1) SIR models are some of the most common and basic epidemic state-based models and (2) our three-dimensional simplex representation of these epidemics emphasizes a "time-invarance" representation of the data (for a refresher see Section **??**).

Model Fitting and Simulations

[Ben says: think about this section and if it highlights that we can bring in outside models...]

After getting a good sense of what a past or current epidemic looks like with EDA, the next step is often model fitting and/or simulations. ~~In this step and the next step (post-processing), we discuss how to easily include external models and simulations originating outside into the~~ **EpiCompare** ~~da~~ this step and the next step (post-proessing), we highlight how practitioners can pair models and simulations of epidemics from outside of **EpiCompare** with analysis and simulation tools in **EpiCompare**.[15] While **EpiCompare** does not focus on fitting model(s) to data, we do provide some flexible functions for simulation of basic discrete-time epidemic-state models. These functions simulation individual-level information based on practitioner estimated transition rates between states and can be combined with `agents_to_aggregate()` to view these simulations through an aggregate lens. The function `simulate_SIR_agents()` simulates a basic SIR epidemic with user inputs for the number of simulations, the initial number in each state, the infection and recovery parameters (β, γ), and the total number of discrete time

this comment but I'm find with the addition...]We say in the intro about highlighting two points of entry into EpiCompare.

[11] [Shannon suggested: "Getting to know" our] [Ben says: even though I was the original one with the quotes - one should always stray away from it. I kept the implicit connection to the section we're in given I'm not sure people are reading the titles super well.]I don't like getting to know without quotes – it's too colloquial. What about 'The practicioner famliarizes herself with data through...'

[12] [Ben says: the action "figuring out" and "groupings" was a bit unclear what actually was going to happen - so I changed it.]fine

[13] [Ben says: I'm reverting to the old text - it's much clearer. - happy to have a a discussion on it. The crossed out text didn't capture what an expert coder was really going to do and how that differed.]fine

[14] [Original text: "By combiing the ideas behind agent2aggregate for three-state models to examine subpopulation trajcetories in 2d simplex space."] [Shannons says: I got rid of the end of this sentence because I think it was putting multiple ideas in one sentence.][Ben says: I'm unclear why this warrants just deleting it. I've provided a newly written sentence...]

[15] [Ben says: the original rewrite didn't seem to clear - specifically in how the practitioner would be using outside models in this step.]fine

steps. This function allows for easy access to SIR model analysis and comparison. Beyond SIR models, the function `simulate_agents()` takes as input a user-specified state-transition matrix and other epidemic parameters to allow the user to create simulations for an outbreak with *any* number of states and any number of transitions among them. This flexibility in states can be used to also reflect group-based dynamics. In turn, this allows for users to explore the space of models in an intuitive way without getting bogged down by too much mathematical detail. For consistency, we have made output from `simulate_agents()` and `simulate_SIR_agents()` compatible with `agents_to_aggregate()` so aggregate information may easily be accessed.

Post-processing

[Ben says: I think we should remind the reader that we care more about simulations, in order to compare fitted models between themselves and the true epidemics.]

[Ben says: this replaces the first paragraph below]

If the practitioner wishes to compare models-to-observations or even models-to-models, they needs to post-process their models and simulations to disseminate the results in an easily digestable format. In general, post-processing of modeling and simulation consists of making summary statistics, plots, and tables. Model ~~The~~ summaries can be very complex, and as a result, a number of epidemic modeling \proglang{R} packages return ~~a~~ special class objects. The special class~~es~~ objects often contain a plethora of information from residuals, model diagnostics, input parameters, and more. While incredibly useful, these special classes can be difficult for novice coders to ~~navigate~~ handle.

To this end, **EpiCompare** provides a series of `fortify`-style methods, called `fortify_aggregate()` which transform output from infectious disease modeling and simulation packages like **pomp** and **EpiModel** into tidy-styled data frames which contain information about the total number of individuals in each state at a given time, for a given simulation. These fortify functions have output that is consistent with that of `agents_to_aggregate()`.

[Shannon says: Here Ben talks about filament compression and tidy_dist mats, etc.] [Ben says: not currently thinking about including tidy_dist stuff - will need to think about this as it is including in the pipeline 2 image...] [16]

To utilize simulations of epidemics in later time-invariant analysis we also provide a function to convert temporally defined epidemics to filamental representations. Specifically, we provide the function `filament_compression()` to convert simulation(s) to filaments as expressed by presenting the epidemic as a ordering of some common fixed number of points so that they are equally spaced along the original path in the proportional state space.

[Shannon says: Shannon's attempt at above paragraph is below.. I tried to more smoothly connect to previous paragraph and then kinda copped out at trying to reword the filament description by just referring back to the previous section.]

EpiCompare also provides a tool to convert time-dependent epidemic simulations into their time-invariant filamental representations. Specifically, `filament_compression()` converts simulations to filaments (see Section **??**) ~~so practitioners can view model simulations and results through a time-invariant lens.~~which allows practitioner to later apply time-invariant tools to these compressed epidemiological objects.[17] These tools were developed to provide

[16] [Ben says: are these comments still relevant?]These comments are resolved

[17] [Ben says: I worry about this statement - it is very broad / seems to claim a lot but not very clearly...]

another natural entry point into the **EpiCompare** data analysis pipeline for situations where modeling and analysis is already completed and practitioners are looking for simple and transparent tools to help understand and disseminate results.

Comparisons and Assessment

Finally, [In **EpiCompare** we provide a set of comparison and assessment tools for models (and model's simulations) that extend beyond the standard performance metrics (e.g. mean squared error or AIC).] Aligned with the discussion in Section **??**, **EpiCompare** provides a set of time-invariant tools to compare and evaluate epidemic models and simulations. We have found these tools to be specifically applicable for situations where only one "season" or "cycle" of the epidemic has occurred (e.g. one flue season).[18]

One tool we provide to assess models is through the creation of geometric prediction regions, useful when we treat epidemics like filaments. If we have a set of simulated epidemics from a model, we can create a geometric prediction region for the expected trajectory of the epdiemice in the state space. [For three-state epidemic models, we provide the `ggplot/ggtern` extension `geom_prediction_band()` which creates a prediction region around the top $1-\alpha$ proportion of the simulations.] In this visual setting, comparing this prediction to the true epidemic trajectory or comparing the prediction regions defined by two different models' simulations can be done visually. [In **EpiCompare** we also provide these prediction regions for epidemic models with with more than three states. The functions `create_convex_hull_structure()` and `create_delta_ball_structure()` create different geometric representations of prediction regions for any dimensional state-based model. For both of these geometric objects, we provide functions to check if a path is contained (`contained()`) and the ability to assess the Hausdorff distance between prediction regions based on simulations from different model (`hassdorf_dist()`).]

[Ben says: this paragraph is replaced by the two above, see "[" segments - they refer to items below.][19] Finally, **EpiCompare** can be used the last step of the data analysis pipeline with its comparison and assessment utility functions. As introduced in Section **??** there's a ~~lot of~~much potential for time-invariant tools to help compare and assess epidemics and models/simulations. In **EpiCompare** we provide a set of comparison and assessment tools for models that extend beyond the standard performance metrics (e.g. mean squared error or AIC) and focus on assessing the structural information the models capture. This approaches work well on models where ~~online~~ one "cycle" of the epidemic has occurred (no recovered individuals have been susceptible again)[20]. Epidemics are complex objects, and we provide tools to create prediction regions with differing desired characteristics[21] from simulated epidemics. For three-state epidemic models, we provide the `ggplot/ggtern` extension `geom_prediction_band()` which creates a prediction region around the top $1-\alpha$ proportion [good place for Ben paper cite?] of the simulations (where the simulations treated as filaments). In **EpiCompare** we also provide these prediction regions for epidemic models with with more than three states. The

[18] [Ben says: Shannon - is this a clear enough use of season / cycle?]

[19] Ben says: Here are some thoughts for the replacment work: (1) The introduction is cleaned up the confusion around "cycle" - hopefully? (2) Shannon's proposed intro wasn't used due to not wanting to claim that this step is always the "last step" of the pipeline. (3) more focus was placed on why we are providing this tools.

[20] [Ben says: Shannon - do you think this is clear / a desirable way to define this - we define it slightly differently in section 2.2.]Shannon says I don't think I understand what you are trying to say there. We should chat about it.

[21] Shannon says: I want to connect epidemics being complex to the fact that prediction is hard and not all prediction regions tell you the same thing

functions `create_convex_hull_structure()` and `create_delta_ball_structure()` create different geometric representations of prediction regions for any state-based model. For both of these geometric objects, we provide functions to check if a path is contained (`contained()`) and the ability to assess the Hausdorff distance between prediction regions based on simulations from different model (`hassdorf_dist()`).

[Ben says: this paragraph is still kept.] We also provide functions to calculate the "extremeness" of a true epidemic trajectory[22] compared to simulated epidemics via the equidistance filamental trajectory representation as mentioned in Section **??**. Specifically, functions like `distance_pseudo_density_function()` can calculate a pseudo-density estimate of the true epidemic relative to simulated ones. Functions `distance_depth_function()` and `local_distance_depth_function()` provide depth scores that suggest how geometrically central an epidemic is to simulations.

4. A tour of EpiCompare

In this section, we highlight many of the tools available in **EpiCompare**. As previously discussed, these tools include data cleaning; visualization; modeling and simulation; post-processing; and comparison and model assessment, in accordance with the data analysis pipeline (Fig. 1). We show a full data analysis from beginning to end that can be accomplished in a streamlined and standardized manner via **EpiCompare**.

4.1. Data and exploratory analysis

We analyze an outbreak of measles in the town of Hagelloch, Germany from 1861-1862, a data set organized by Pfeilsticker (1863). The data was later made visible by Oesterle (1992) and made available in an R by Meyer *et al.* (2017). The Hagelloch data includes a rich set of features including household members, school level, household locations, date of first symptoms (prodromes), date of measles rash, and even the alleged infector. A subset of the data is shown in Table 1. Because of these rich features, this data set has been an ideal testing ground methodology in infectious disease epidemiology and is used in work by Neal and Roberts (2004); Britton *et al.* (2011); Groendyke *et al.* (2012); Becker *et al.* (2016).

Table 1: Subset of Hagelloch infection data. Features include the person ID, household ID (HH ID), age, sex, class level (Pre-K/1st/2nd), date of first symptoms, date of the appearance of the measles rash, and the alleged infector ID of the individual.

ID	HH ID	Name	Age	Sex	Class	Symp. Start	Rash Date	Infector ID
1	61	Mueller	7	female	1st class	1861-11-21	1861-11-25	45
2	61	Mueller	6	female	1st class	1861-11-23	1861-11-27	45
3	61	Mueller	4	female	preschool	1861-11-28	1861-12-02	172
4	62	Seibold	13	male	2nd class	1861-11-27	1861-11-28	180
5	63	Motzer	8	female	1st class	1861-11-22	1861-11-27	45
45	51	Goehring	7	male	1st class	1861-11-11	1861-11-13	184

[22] Ben says: accepted.

With **EpiCompare**, we can easily obtain the empirical cumulative incidence function with respect to the measles rash appearance (variable `ERU`) with the following tidy-style function, `agents_to_aggregate()`. The function `agents_to_aggregate()` is a key component of **EpiCompare**, allowing the user to easily switch from an individual-level (i.e. an agent) view of a disease to an aggregate level. For example, the below code shows how we can convert the agent data to a cumulative incidence of the measles rash, in order to see how the disease spread through the population over time. We can then compare the cumulative incidence of the rash to the cumulative incidence of the prodromes, i.e. the initial symptoms. We do this with the below code, and a part of the cumulative incidence data output is shown in Table 2. The argument `integer_time_expansion` indicates whether we should include all time points in the recorded range of the data or only when there is a change in the incidence.

```
R> cif_rash  <- hagelloch_raw %>%
+   mutate(time_of_rash = as.numeric(ERU - min(PRO, na.rm = TRUE))) %>%
+   agents_to_aggregate(states = time_of_rash,
+                       integer_time_expansion = FALSE) %>%
+   mutate(type = "Rash")
```

Table 2: Turning the individual-level information from the Hagelloch data to an aggregate view of the cumulative incidence of the measles rash in the population over time.

Time	# Susceptible	# Total rash appearances
0	188	0
4	187	1
7	186	2
9	185	3
12	183	5

One possible question of interest is the duration between initial onset of prodromes and the appearance of the measles rash. Since `agent_to_aggregate()` outputs a tidy-style data frame, it is a simple task to plot the two sets of incidence curves on the same graph (Fig. 5).

```
R> cif_prodromes <- hagelloch_raw %>%
+   mutate(time_of_PRO = as.numeric(PRO - min(PRO, na.rm = TRUE))) %>%
+   agents_to_aggregate(states = time_of_PRO,
+                       integer_time_expansion = FALSE) %>%
+   mutate(type = "Pro")

R> plot_df <- bind_rows(cif_rash, cif_prodromes)
R>
R> ggplot(data = plot_df,
+         aes(x = t, y = X1, col = type)) +
+   geom_step() +
+   labs(title = "Cumulative incidence of measles appearance",
+         x = "Time (days relative to first prodrome appearance)",
```

```
+         y = "Cumulative incidence of event") +
+    coord_cartesian(xlim = c(0, 55)) +
+    scale_color_manual(values = c("blue", "red"))
```

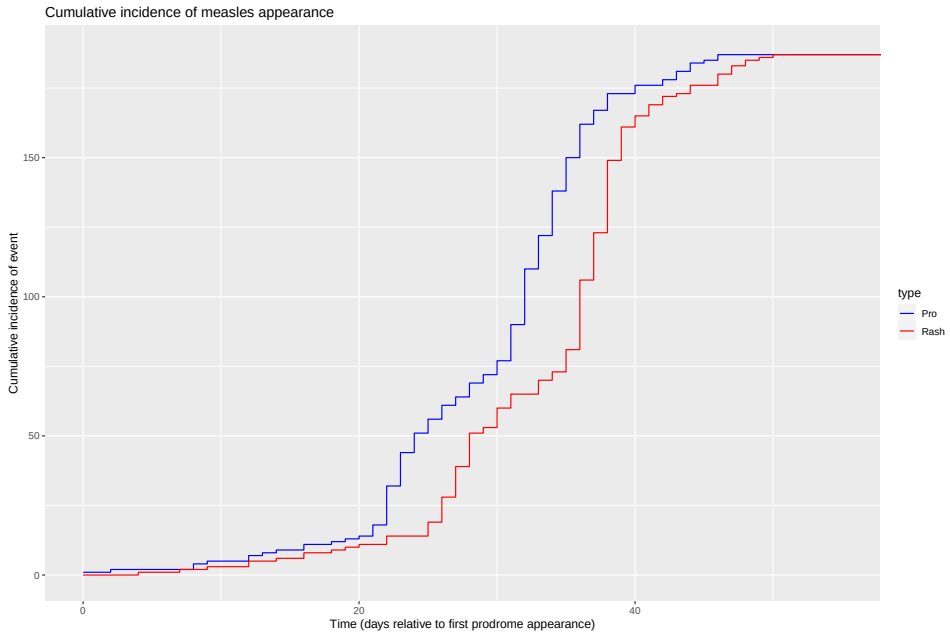

Figure 5: Empirical cumulative incidence functions of prodrome (symptom) onset and measles rash appearance. We see that there is approximately a a constant lag between the two curves.

The real power of `agents_to_aggregate()` lies in its ability to aggregate over any number of pre-specified states. For example, the Hagelloch data sets contains two columns, `tI` and `tR`, the time of infection and recovery, respectively of each individual. We can then plot the SIR values through a time-invariant lens using **ggplot2** and **ggtern** functions (as shown in Fig. 6) or with our custom `geom`, `geom_aggregate`, which takes the raw agent data as input.

```
R> hagelloch_sir <- hagelloch_raw %>%
+    agents_to_aggregate(states = c(tI, tR),
+                        min_max_time = c(0, 55)) %>%
+    rename(time = t, S = X0, I = X1, R = X2)
R>
R>
R> ggplot(hagelloch_sir, aes(x = S, y = I, z = R))+
+    coord_tern() +
+    geom_path() +
+    labs(x = "S", y = "I", z = "R",
+         title = "Time invariant view of Hagelloch measles outbreak") +
+    theme_sir(base_size = 24)
```

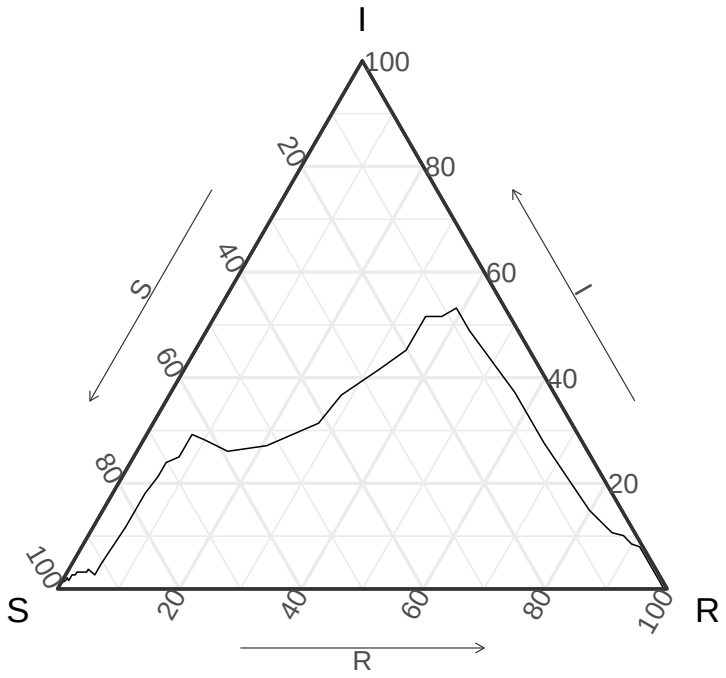

Figure 6: Time invariant view of the Hagelloch epidemic where we view the individuals in Susceptible, Infectious, or Recovered states. We see there are two peaks of infection (the vertical axis).

Moreover, we can look at the outbreaks of the disease by group within `agent_to_aggregate()` or `geom_aggregate()`. This allows us to examine differences among the different groups of individuals. For example, we show the time invariant outbreak by class level in Figure 7. Immediately, we see that time invariant infection curve is different for the pre-school class compared to the 1st class. In the 1st class, we see about 95% of the class become infected and less than 10% of them having recovered, which may be indicative of a super-spreading event. This suspicion is further confirmed in that 26 of the 30 1st class students have been reportedly infected by the same individual.

```
R> hagelloch_raw %>%
+   ggplot(aes(y = tI, z = tR, color = CL)) +
+   geom_aggregate(size = 2) + coord_tern() +
+   labs(x = "S", y = "I", z = "R",
+        color = "Class") +
+   scale_color_brewer(palette = "Dark2") +
+   facet_wrap(~CL)
```

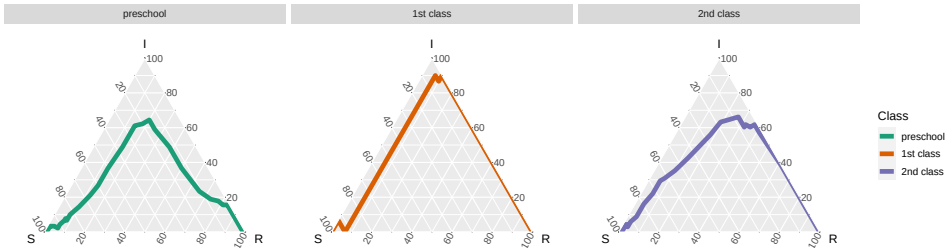

Figure 7: Time invariant outbreak curves for the three class groups. The pre-school class has a distinct peak of infection whereas the peak infection point for the other two classes are less well defined.

Along with multiple epidemic states, the function `agents_to_aggregate` can also be extended to populations with vital dynamics (e.g. birth and death) and examples of this are shown in the package vignette. In summary, `agents_to_aggregate()` is a multi-purpose workhorse that may be leveraged to convert individual level records into aggregate information that may be more useful for some forms of epidemic modeling such as compartment modeling.

Up to this point, we have used **EpiCompare** in the context of observed data. We also want to compare statistical models, and **EpiCompare** aids in that process via a simple yet flexible individual-level simulator, conversion tools for popular epidemic model packages, and model assessments. We demonstrate an example here.

We first try to model the Hagelloch data with a stochastic SIR model, which we refer to as the 'simple SIR.' In our vignette, we show how to fit this simple SIR model via maximum likelihood and simulate from the model with those best fit parameters. Our function `simulate_agents()` generates individual level data according to discrete time multinomial draws, which depend on the number of individuals in each state at the previous time step and a matrix of transition probabilities. For example, the below code generates 100 simulations of an outbreak of a disease with one initial infector in a population of $n = 188$ individuals.

```
R> trans_mat <- matrix(c("X0 * (1 - X1 * par1 / N)", "X0 * X1  * par1 / N", "0",
+                   "0", "X1 * (1 - par2)", "par2 * X1",
+                   "0", "0", "X2"), byrow = TRUE, nrow = 3)

R> set.seed(2020)
R>
R> best_params <- c("beta" = .36, "gamma" = .13)
R> ## This is the SIR representation
R>
R> rownames(trans_mat) <- c("S", "I", "R")
R> init_vals <- c(187, 1, 0)
R> par_vals <- c(par1 = best_params[1], par2 = best_params[2])
R> max_T <- 55
R> n_sims <- 100
R>
R> agents <- simulate_agents(trans_mat,
+                         init_vals,
+                         par_vals,
```

```
+                          max_T,
+                          n_sims,
+                          verbose = FALSE)

R> agg_model <- agents %>% group_by(sim) %>%
+   agents_to_aggregate(states = c(I, R)) %>%
+   mutate(Type = "Simple SIR")
```

The result of our simulation is the object `agents` which is a 18800 $\times$ 5 data frame, which details the time of entry into the S, I, and R states for a given simulation. Before we examine the results of this simple SIR model, we will also examine another, more sophisticated SIR model, this time from the package **EpiModel**. Briefly, this model first fits a contact network to the set of indivduals, where the class of the student is a covariate. The model then simulates a SIR-epidemic on that network.

```
R> library(EpiModel)
R> ## WARNING:  Will take a minute or two
R>
R> set.seed(42)
R> nw <- network.initialize(n = 188, directed = FALSE)
R> nw <- set.vertex.attribute(nw, "group", rep(0:2, each = 90, 30, 68))
R> formation <- ~edges + nodematch("group") + concurrent
R> target.stats <- c(200, 300, 200)
R> coef.diss <- dissolution_coefs(dissolution = ~offset(edges),  duration = 5)
R> est1 <- netest(nw, formation, target.stats, coef.diss, edapprox = TRUE)
R>
R> param <- param.net(inf.prob = 0.1, act.rate = 5, rec.rate = 0.1)
R> status.vector <- c(rep(0, 90), rep(0, 30), rep(0, 67), 1)
R> status.vector <- ifelse(status.vector == 1, "i", "s")
R> init <- init.net(status.vector = status.vector)
R> control <- control.net(type = "SIR", nsteps = 55,
+                          nsims = 100, epi.by = "group")
R> epimodel_sir <- netsim(est1, param, init, control)
```

The output of this model is `epimodel_sir`, an object of class `netsim`, which contains a plethora of modeling information. We provide the function `fortify_aggregate()`, which can take objects from specialized classes of modeling output and transform it into a tidy-style data frame.

```
R> fortified_net <- fortify_aggregate(epimodel_sir,
+                                     states = c("s.num", "i.num", "r.num")) %>%
+   mutate(Type = "EpiModel SIR",
+          sim = as.numeric(gsub("sim", "", sim)))
```

We can then analyze the results of the two models side by side as time-invariant epidemic curves. The results are shown in Figure 8, where a 90% prediction band is estimated from

the delta ball method for each of the two models. For the Simple SIR model, we see that the data generally covers the data fairly well but clearly misses the second peak of infection. We also see that the prediction band is very large, covering up a large area of the ternary plot. On the other hand, for the **EpiModel** model, we see that the prediction band covers the data quite well and takes up less area.

```
R> both_models <- bind_rows(agg_model, fortified_net)
R>
R>
R> g <- ggplot() + geom_prediction_band(data = both_models %>% filter(t != 0),
+          aes(x = X0, y = X1, z = X2,
+               sim_group = sim, fill = Type),
+          alpha = .5,
+          conf_level = .90)
```

```
R> g +   geom_path(data = both_models %>% filter(t !=0),
+             aes(x = X0, y = X1, z = X2, group = paste(Type, sim)),
+             alpha = .3, col = "gray40") +
+     coord_tern() + theme_sir(base_size = 24) +
+   geom_point(data = hagelloch_sir,
+              aes(x = S, y = I, z =R), col = "black") +
+   labs(title = "Simple SIR model",
+        subtitle = "90% Prediction band and original data",
+        x = "S", y = "I", z = "R") +
+   scale_fill_manual(values = c("#006677", "#AA6600")) +
+   facet_wrap(~Type) +
+   theme(legend.position = "bottom")
```

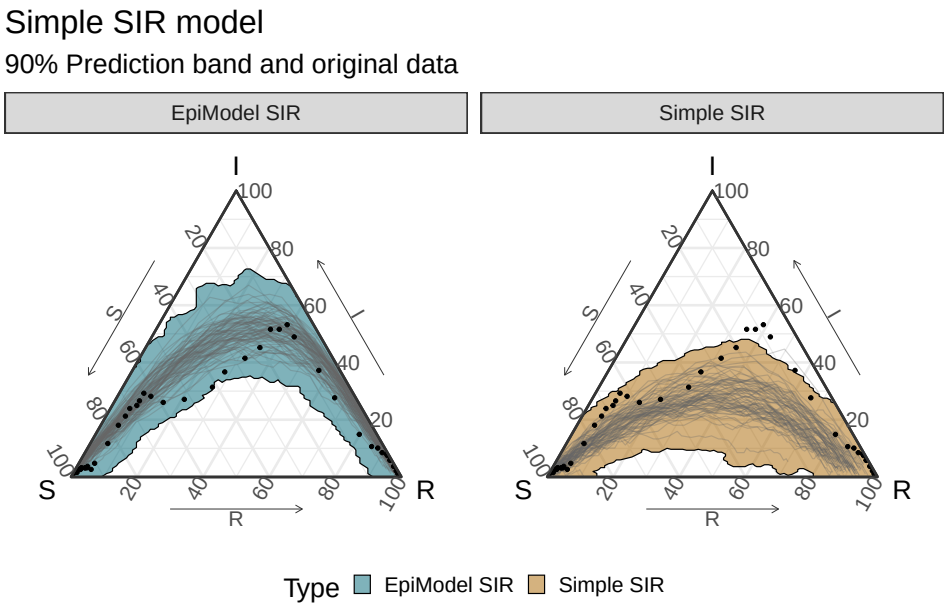

Figure 8: Original Hagelloch SIR data (black) along with 90% prediction band and actual simulation paths from the Simple SIR and the EpiModel SIR models.

However, both models are not a good fit to the filamental path as opposed to the individual points in (S, I, R)-space. This can be captured with the set of simulations both models predict (gray lines), which all generally have a single defined peak of infection whereas the data certainly looks like it has two distinct peaks, likely caused by our assumed super-spreader event. This observation is backed up by the below analysis that demonstrates that the estimated pseudo-density of the observed epidemic (relative to the simulations from either model) is much less likely then **any** of the simulations (reported in Table 4). In conclusion, **EpiCompare** makes it clear that, at a glance, 1) the **EpiModel** network model is a better fit than the Simple SIR model, and 2) the fit is only good at the geometric filamental level as opposed to the epidemic trajectory filamental level.

```
R> #-- after cleaning up and combining --
R> all_together_df <- rbind(simple_sir,
+                           hagelloch_sir2)
```

Table 3: Top and bottom 2 rows of `all_together_df, combining both simulated epidemics and the tr`

Type	sim	t	S	I	R
Simple SIR	1	0	188	0	0
Simple SIR	1	1	187	1	0
true observation	0	54	1	0	187
true observation	0	55	1	0	187

```
R> compression_df <- all_together_df %>% group_by(Type, sim) %>%
```

```
+   filament_compression(data_columns = c("S","I","R"),
+                        number_points = 20)


R> tdmat <- compression_df %>%
+   dist_matrix_innersq_direction(
+     position = c(1:length(compression_df))[
+       names(compression_df) %in% c("S","I", "R")],
+     tdm_out = T)
R>
R> simple_sir_true_obs_info <- tdmat %>%
+   compare_new_to_rest_via_distance(
+     new_name_id = data.frame(Type = "true observation", sim = 0),
+     distance_func = distance_psuedo_density_function,
+     sigma = "20%")
```

Table 4: The extremeness of the true simulations based on comparing psuedo-density estimates between true vs simulated curves

Type	simulations-based estimated psuedo-density	proportion of simulations with lower estimated psuedo-density
Simple SIR	0.0117621	0.01
EpiModel SIR	0.0356503	0.01

Overall, **EpiCompare** aids in the data analysis pipeline for both novice and expert practitioners and coders alike. These tools encourage model and simulation exploration of many of the existing and well-supported packages that already exist, and side-by-side comparison thereof. Finally, we hope that practicioners will consider using time-invariant analysis when trying to assess and compare epidemics and epidemic models.

A. Appendix

A.1 Proof of Theorem 1

Proof. Harko *et al.* (2014) provide an analytical solution for the Kermack and McKendrick equations (Eq. (1)) by reparameterizing the ODEs so that $\mathcal{S}(u) = S(t)$, $\mathcal{I}(u) = S(t)$, and $\mathcal{R}(u) = R(t)$ for $0 < u_T < 1$ with

$$\begin{aligned} \mathcal{S}(u) &= S(0)u \\ \mathcal{I}(u) &= N - R(0) + NR_0^{-1} \log u - S(0)u \\ \mathcal{R}(u) &= R(0) - NR_0^{-1} \log u, \end{aligned} \tag{2}$$

and u and t are related by the following integral,

$$\begin{aligned} t &= \int_u^1 \frac{N}{\beta\tau(N - R(0) + R_0^{-1}\log\tau - S(0)\tau)} d\tau \\ &= \int_u^1 \frac{1}{\beta f(S(0), R(0), N, R_0, \tau)} d\tau \\ &= \int_u^1 \frac{1}{\beta f(\tau)} d\tau, \end{aligned}$$

where we have made the denominator of the integral a function of N, the initial values, R_0, and τ, which we further condense to $f(\tau)$ for brevity. Then for a given t we want to find s such that $(S_1(t), I_1(t), R_1(t)) = (S_2(s), I_2(s), R_2(s))$. Or equivalently, for a fixed u want to find v such that $\mathcal{S}_1(u) = \mathcal{S}_2(v)$ and then the corresponding t and s are given by

$$\begin{aligned} t &= \int_u^1 \frac{1}{\beta_1 f(\tau)} d\tau \\ s &= \int_v^1 \frac{1}{\beta_2 f(\tau)} d\tau. \end{aligned}$$

Note that since the equations in Eq. (2) are functions of the initial values and R_0, then $u = v$. We then can find a relation for s,

$$\begin{aligned} s &= \int_u^1 \frac{1}{\beta_2 f(\tau)} d\tau \\ &= \int_u^1 \frac{1}{a\beta_1 f(\tau)} d\tau \\ &= \frac{1}{a}\int_u^1 \frac{1}{\beta_1 f(\tau)} d\tau \\ &= \frac{1}{a} t. \end{aligned}$$

□

References

Anderson RM, May RM (1992). *Infectious diseases of humans: dynamics and control*. Oxford university press.

Becker AD, Birger RB, Teillant A, Gastanaduy PA, Wallace GS, Grenfell BT (2016). "Estimating enhanced prevaccination measles transmission hotspots in the context of cross-scale dynamics." *Proceedings of the National Academy of Sciences*, **113**(51), 14595–14600.

Biggerstaff M, Alper D, Dredze M, Fox S, Fung ICH, Hickmann KS, Lewis B, Rosenfeld R, Shaman J, Tsou MH, Velardi P, Vespignani A, Finelli L, Chandra P, Kaup H, Krishnan R, Madhavan S, Markar A, Pashley B, Paul M, Meyers LA, Eggo R, Henderson J, Ramakrishnan A, Scott J, Singh B, Srinivasan R, Bakach I, Hao Y, Schaible BJ, Sexton JK, Del

Valle SY, Deshpande A, Fairchild G, Generous N, Priedhorsky R, Hickman KS, Hyman JM, Brooks L, Farrow D, Hyun S, Tibshirani RJ, Yang W, Allen C, Aslam A, Nagel A, Stilo G, Basagni S, Zhang Q, Perra N, Chakraborty P, Butler P, Khadivi P, Ramakrishnan N, Chen J, Barrett C, Bisset K, Eubank S, Anil Kumar VS, Laskowski K, Lum K, Marathe M, Aman S, Brownstein JS, Goldstein E, Lipsitch M, Mekaru SR, Nsoesie EO, Gesualdo F, Tozzi AE, Broniatowski D, Karspeck A, Tse ZTH, Ying Y, Gambhir M, Scarpino S (2016). "Results from the centers for disease control and prevention's predict the 2013-2014 Influenza Season Challenge." *BMC Infectious Diseases*, **16**(1), 1–10. ISSN 14712334. doi:10.1186/s12879-016-1669-x. URL http://dx.doi.org/10.1186/s12879-016-1669-x.

Britton T, Kypraios T, O'Neill PD (2011). "Inference for epidemics with three levels of mixing: methodology and application to a measles outbreak." *Scandinavian Journal of Statistics*, **38**(3), 578–599.

CDC (2021). "CDC COVID Data Tracker." URL https://covid.cdc.gov/covid-data-tracker/#cases_casesper100klast7days.

Ciollaro M, Genovese CR, Wang D (2016). "Nonparametric clustering of functional data using pseudo-densities." *Electronic Journal of Statistics*, **10**(2), 2922–2972. ISSN 19357524. doi:10.1214/16-EJS1198.

Dong E, Du H, Gardner L (2020). "An interactive web-based dashboard to track COVID-19 in real time." *The Lancet infectious diseases*, **20**(5), 533–534.

Ferguson N, Laydon D, Nedjati Gilani G, Imai N, Ainslie K, Baguelin M, Bhatia S, Boonyasiri A, Cucunuba Perez Z, Cuomo-Dannenburg G, *et al.* (2020). "Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand."

Gallagher S, Chang A, Eddy WF (2020). "Exploring the nuances of R0: Eight estimates and application to 2009 pandemic influenza." *arXiv preprint arXiv:2003.10442*.

Geenens G, Nieto-Reyes A (2017). "On the functional distance-based depth."

Groendyke C, Welch D, Hunter DR (2012). "A network-based analysis of the 1861 Hagelloch measles data." *Biometrics*, **68**(3), 755–765.

Hamilton NE, Ferry M (2018). "ggtern: Ternary Diagrams Using ggplot2." *Journal of Statistical Software, Code Snippets*, **87**(3), 1–17. doi:10.18637/jss.v087.c03.

Harko T, Lobo FS, Mak MK (2014). "Exact analytical solutions of the Susceptible-Infected-Recovered (SIR) epidemic model and of the SIR model with equal death and birth rates." *Applied Mathematics and Computation*, **236**, 184–194. ISSN 00963003. doi:10.1016/j.amc.2014.03.030. 1403.2160, URL http://dx.doi.org/10.1016/j.amc.2014.03.030.

Jenness SM, Goodreau SM, Morris M (2018). "EpiModel: An R Package for Mathematical Modeling of Infectious Disease over Networks." *Journal of Statistical Software*. doi:10.18637/jss.v084.i08.EpiModel.

Kermack WO, McKendrick AG (1927). "A contribution to the mathematical theory of epidemics." *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, **115**(772), 700–721.

King AA, Nguyen D, Ionides EL (2016). "Statistical inference for partially observed markov processes via the R package pomp." *Journal of Statistical Software*, **69**(12), 1–43. ISSN 15487660. doi:10.18637/jss.v069.i12. 1509.00503.

Meyer S, Held L, Höhle M (2017). "Spatio-Temporal Analysis of Epidemic Phenomena Using the R Package surveillance." *Journal of Statistical Software*, **77**(11), 1–55. doi:10.18637/jss.v077.i11.

MIDAS Network (2021). "Online Portal for COVID-19 Modeling and Research." URL https://midasnetwork.us/covid-19/.

Neal PJ, Roberts GO (2004). "Statistical inference and model selection for the 1861 Hagelloch measles epidemic." *Biostatistics*, **5**(2), 249–261. ISSN 14654644. doi:10.1093/biostatistics/5.2.249.

Oesterle H (1992). "Statistische Reanalyse einer Masernepidemie 1861 in Hagelloch."

Pfeilsticker A (1863). "Beiträge zur Pathologie der Masern mit besonderer Berücksichtigung der statistischen Verhältnisse." URL http://www.archive.org/details/beitrgezurpatho00pfeigoog.

Rvachev LA, Longini IM (1985). "A mathematical model for the global spread of influenza." *Mathematical Biosciences*, **75**(1), 3 – 22. ISSN 0025-5564. doi:http://dx.doi.org/10.1016/0025-5564(85)90064-1. URL http://www.sciencedirect.com/science/article/pii/0025556485900641.

The Washington Post (2021). "Coronavirus US Cases and." URL https://washingtonpost.com/graphics/2020/national/coronavirus-us-cases-deaths/.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." *Journal of Open Source Software*, **4**(43), 1686. doi:10.21105/joss.01686.

Worby CJ, Chaves SS, Wallinga J, Lipsitch M, Finelli L, Goldstein E (2015). "On the relative role of different age groups in influenza epidemics." *Epidemics*, **13**, 10–16.

Affiliation:

Shannon K. Gallagher
Biostatistics Research Branch
National Institute of Allergy
and Infectious Diseases
5603 Fishers Lane
Rockville, MD 20852
E-mail: shannon.gallagher@nih.gov
URL: http://skgallagher.github.io

Benjamin LeRoy
Dept. of Statistics & Data Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
E-mail: bpleroy@andrew.cmu.edu
URL: https://benjaminleroy.github.io/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd