Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Lindsay Hexter April 15, 2018

Monkeying Around: Automatically Analyzing Malaria Infections in Rhesus Macaques

by

Lindsay Hexter

Jinho D. Choi, Ph.D.
Adviser

Department of Mathematics and Computer Science

Jinho D. Choi, Ph.D.
Adviser

Arri Eisen, Ph.D.
Committee Member

Davide Fossati, Ph.D.
Committee Member

Mary R. Galinski, Ph.D.
Committee Member

Astrid Prinz, Ph.D.
Committee Member

2018

Monkeying Around: Automatically Analyzing Malaria Infections in Rhesus Macaques

by

Lindsay Hexter

Jinho D. Choi, Ph.D.

Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2018

Abstract

Monkeying Around: Automatically Analyzing Malaria Infections in Rhesus Macaques

By Lindsay Hexter

In today's age of big data, automatic processing techniques are becoming more important than ever, especially in the field of biology and medicine. Many studies focus on genomic data, following the rise of high throughput sequencing; this project instead analyzes certain blood data parameters taken from rhesus macaques housed in Yerkes National Primate Research Center at Emory University.

The Joyner et al. 2016 paper, "*Plasmodium cynomolgi* infections in rhesus macaques display clinical and parasitological features pertinent to modelling vivax malaria pathology and relapse infections," was the initial motivation for this study (Joyner et al., 2016). Joyner and his team follow the infection of malaria species *P.cynomolgi* in monkeys, taking blood data and other biological information daily. While the paper discusses possible points of difference between monkeys of varying disease severity, we endeavored to find an automatic way to use these "clinical and parasitological features" to characterize and predict aspects of malaria, including severity and stage of the longitudinal infection.

We propose to replicate existing analyses and to add new insights via various computational techniques. Machine learning is traditionally used for very large datasets, and thus this thesis intends to provide a proof of concept for automatically analyzing these types of smaller datasets, given restrictions studying monkeys. The flow of computation is as follows: normalization of data, creation of mathematical models, residual calculation, formation of residual matrices for clustering, and lastly the generation of regression models. The aforementioned procedure is then applied to shifted data for comparison, using Bayesian optimization. This study therefore

provides a comprehensive framework for automatic analysis of medical data, which can be applied to other datasets.

Monkeying Around: Automatically Analyzing Malaria Infections in Rhesus Macaques

by

Lindsay Hexter

Jinho D. Choi, Ph.D.

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2018

Acknowledgements

Thank you to all my committee members who have supported me throughout my career at Emory, and a special thanks to Dr. Choi who has pushed me to achieve greater things since my first course with him in 2016! Also thank you to researchers from the Malaria Host-Pathogen Interaction Center (MaHPIC) team, Dr. Joyner and Dr. Galinski, whose guidance was valuable in understanding the experiments I analyzed. Last but not least, a huge thanks to Spiller Park coffee shop for fueling my study sessions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of medically-related research is characterized by finding models that are similar enough to humans to uncover insights that will apply to our needs. Moreover, finding a model system is a balance between similarity and practicality; while monkeys are very similar to humans, the number of instances in experiments with these animals is necessarily small. It is nonetheless important to study diseases in monkeys, as done at the Yerkes National Primate Research Center (YNPRC) at Emory University, since they are evolutionarily close to humans.

The purpose of the background paper to this thesis by Joyner et al. was to assess certain parameters important to the development and outcome of the given *Plasmodium* infection, while providing data on which other experiments could build (Joyner et al., 2016). Even as there are few monkeys in the study, the dimensionality of the data for each monkey is large, hence motivating the use of automatic analysis. However, these datasets provide a challenge not only to manual analysis but also to automatic analysis, as few instances and many attributes characterize

the shape of the data (rather than vice versa, with many instances and fewer attributes, as many machine learning datasets are).

The approach taken in this study is a combination of tactics used in machine learning, applied specifically to medical research, which includes clustering, curve fitting, and regression modeling. Because the data has many dimensions, these techniques can help to uncover insight that cannot be found manually. The small size of the dataset confirms this project as a proof of concept that can be built upon and fine-tuned with more data and initial manual validation, rather than as a traditional machine learning approach.

1.1 Thesis Statement

We intend to provide a general framework for automatic analysis of small biological datasets, using experiments conducted by Joyner et al. at the YNPRC as examples. While the initial structure is based on the 2016 paper, "*Plasmodium cynomolgi* infections in rhesus macaques display clinical and parasitological features pertinent to modelling vivax malaria pathology and relapse infections" (Joyner et al., 2016), the same framework can be applied to other experiments from the MaHPIC team as well as to other studies with similar biological time series data. Therefore, this project provides an unconventional use of certain machine learning techniques on smaller datasets to determine a procedure for analyzing medical data; which consists of normalization, mathematical modeling, residual calculation, residual matrices, and regression modeling. Moreover, Bayesian optimization is not generally used for this type of analysis, and so it is implemented to provide yet another layer of insight to the differences among the monkey data studied here.

Chapter 2

Background

Many areas of the sciences benefit from automatic analysis, and thus foundations are established for different steps in data processing and analyzing. While the problem explored in this project is different in that it combines ideas from computational modeling fields and machine learning studies, previous work can be used as a starting point. Moreover, motivation behind this thesis still echoes that of medically-related projects in general- a better way of analyzing data can help reduce time and monetary costs while promoting goals to find better treatments.

2.1 Why Malaria

The impetus for studying malaria is shown by its public health cost, both in lives and dollars; 3.2 billion people worldwide are at risk (CDC, n.d.), and the US alone spent $1 billion in 2016, contributing to the global total of $2.7 billion (WHO, 2017). Therefore, researching the course of the infection to help derive better treatments is crucial.

Malaria microorganisms are comprised of over 100 species, each having different clinical implications and affecting diverse animal species. This thesis focuses most specifically on the course of *P.cynomolgi*, which is closely related to the human parasite *P.vivax*. The dormant liver populations of *P.vivax*, hypnozoites, can make this parasite hard to diagnose, as many patients that are still contagious do not receive treatment for lack of symptoms (CDC, n.d.). These dormant stages necessitate combination treatments because of varying implications on transmission, infection severity, and relapse presented at different stages (Baird et al., 2016).

It is pertinent to study similar malaria-inducing species in model organisms because as stated, relapse stages in *P.vivax* make this parasite difficult to diagnose and treat; for this reason, *P.cynomolgi* was studied at YNPRC in rhesus macaques. *P.cynomolgi* is phylogenetically similar to *P.vivax*, and they both are characterized by the aforementioned hypnozoites (Sanger Institute, n.d.). In the Joyner et al. 2016 study (Joyner et al., 2016), the course of *P.cynomolgi* was followed in five rhesus macaques. Monkeys are evolutionarily close to humans and show similar infection characteristics, and thus providing specific data on malaria helps to formulate treatments and facilitate understanding of human infection. The first step in this research is gathering data to study the infection in a more controlled environment, as done at YNPRC. However, finding ways to quickly analyze these results allows for faster and more in-depth comprehension of the given

experiment, influencing the direction of future studies and contributing to overall understanding of the infection. This globally pressing public health issue thus motivates the use of technology in analyzing the results of malaria experiments.

2.2 Preprocessing and Normalization of Data

With biological data, it is common to have very large or missing values; therefore, a common preprocessing technique is to fill any missing values with either zero or the row average of the data and subsequently take the log to smooth any extreme values (Gomez, 2010). I used this technique to compensate for missing and/ or extreme data-points.

In many machine learning studies, normalization is used to "equalize ranges of the features and make them have approximately the same effect in the computation of similarity" (Aksoy & Haralick, 2000). The reason for doing so is that common distance metrics are affected by magnitude; therefore, in terms of algorithms like clustering, the similarity between two points can be skewed without normalization. One such metric is Euclidean distance, which defines the distance between two points q and p, in n-dimensional space, as:

$$d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{2.1}$$

With distance metrics like Euclidean, normalization then prevents skewing of data with respect to magnitude as shown in the formula. In general, normalization is important in this study because of the vast variation encountered in biological data (Gomez, 2010).

Normalizing data to the interval [0, 1] can be performed using various formulas. Two very common ways are to divide each point by the maximum value as in 2.2 or to subtract the

minimum from each point and divide by the difference between the maximum and minimum values, as in 2.3 (Witten et al., 2017). The latter method scales each value to the interval [0,1], and so I used this method for creating residual matrices via `sklearn MinMaxScaler` as discussed in Chapter 3. `sklearn` is a library in Python used for data processing and machine learning.

$$x_i = \frac{x_i}{max(values)} \tag{2.2}$$

$$x_i = \frac{x_i - min(values)}{max(values) - min(values)} \tag{2.3}$$

Additionally, while these formulas scale the data in each point to the interval [0, 1], I also chose to normalize each dataset as a unit vector, following *l1*-normalization as in 2.4 (Stern et al., 2007); this ensures that data for any clinical parameter and any monkey are in the same range and creates more discrepancy between very small and very large values (to emphasize time-points that could be more important).

$$x_i = \frac{x_i}{\sum_{j=1}^{n} x_j} \tag{2.4}$$

There are existing methods in `sklearn` to do so; however, in order to best fit to the datatypes I created, I chose not to use the method from `sklearn` and instead implement my own, following the formula in 2.4.

2.3 Nonlinear Least Squares and Gaussian Fitting

Curve fitting is used extensively to calculate and predict physiological values in various bio-related fields, and in order to do so, some loss function must be reduced to find optimal curve parameters. A common algorithm used is the Levenberg-Marquardt method, which avoids brute-force calculation of all possible parameter values and combinations (Ahearn et al., 2005).

`curve_fit` implements the Levenberg-Marquardt algorithm for solving nonlinear least squares. This algorithm combines gradient descent, a way to guess parameters for minimizing error rather than solving complex derivatives, and the Gauss-Newton method, where optimal parameters are assumed to be quadratic to simplify derivative calculation (derivative = 0 signifies minimal loss). The Levenberg-Marquardt algorithm alternates between these two methods, depending on how close the parameters are from the optimal solution (as defined by the loss function) (Gavin, 2011). Fitting mathematical functions to biological data is important for later analysis steps, and so I used the `curve_fit` function from `sklearn` to fit Gaussian functions for each data section found (as described in 3.2).

2.4 Regression Modeling

While correlation coefficients describe the relationship between variables, regression modeling is instead a way to predict some Y based on some X. This allows for a more fine-grained analysis of relationships among multiple variables at a time to identify which are best predictors (i.e. hold the most 'weight'). Finding a model is important for data simulation and prediction, and it provides for doing so in the simplest way possible (Motulsky & Christopoulos, 2004). I chose

linear regression in order to find the best linear combination of all clinical variables that could predict the number of parasites per microliter, a good indicator of disease severity.

2.4.1 Ridge Regression

Linear regression can lead to over-fitting if there are many free parameters (Ng, 2004); over-fitting signifies that while the model predicts the training data very well, it results in high-error predictions for new samples that are different from this training data. Because the predictive X was high-dimensional, using a method to avoid a highly specific and un-generalizable model was necessary. I used Ridge regression from `sklearn` to include *l2*-regularization, since this method helps to prevent over-fitting of certain coefficients. *l2*-regularization adds the squared sum of weights w as a regularizer in updating coefficients shown in 2.5, penalizing a likely fit that has improbable coefficients (the larger the coefficients, the less likely they will be chosen). In this equation, there are k coefficients (k clinical parameters) and n days to predict; therefore, the cost is greater if all coefficients are large (offset by some parameter *lambda*), and so this regularization favors weights evenly distributed across parameters (rather than sparse weights).

$$cost = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{k} x_{ij} w_j^2) + \lambda \sum_{j=1}^{k} w_j^2 \tag{2.5}$$

Even as *l1*-regularization is stated to beat *l2*-regularization in work by Andrew Ng (Ng, 2004), in practice it is less accurate and slower as identified in the LIBLINEAR documentation. As seen in Table 2.1, the larger coefficients originally obtained with a standard linear model are penalized after application of Ridge regression.

TABLE 2.1: RMe14: Weights for all features, example comparing standard Linear Regression and Ridge regression (highlighted weights were especially penalized in the Ridge model).

	Feature	Ridge	Standard Model
0	gran	3.61	2.89
1	hct	-4.75	-7.00
2	hgb	-4.23	-64.96
3	lymph	1.96	2.8
4	mch	-2.58	103.15
5	mchc	2.22	-83.40
6	mcv	-6.97	-80.02
7	mono	2.24	2.56
8	mpv	-15.48	-19.07
9	plt	2.75	-2.27
10	rbc	12.39	81.41
11	rdw	-21.60	-25.86
12	# ret	-1.89	-0.90
13	ret / uL	1.96	0.97
14	ret %	0.31	0.33
15	wbc	-5.42	-6.75

2.5 Clustering

Unsupervised learning is used for problems in which classifiers or labels are generally unknown for entities in data; instead, similar entities are grouped automatically to establish connections among those clusters. Clustering is extremely useful for gleaning information about unlabeled data, and in the case of certain clinical parameters in this study, it is interesting to see which of the parameters cluster together to derive certain classes. There are various types of clustering algorithms, and many were tested in this project to determine which would be best suited based on evaluation criteria.

Many approaches exist to implement clustering, all employing different methods to characterize entities based on some similarity metric. The best-performing algorithm in this project as determined by silhouette score was *kmeans*, a partition-based algorithm. *Kmeans* begins with initial k centroids, which may be chosen randomly or based on furthest distance. Until some given stopping criteria, such as when cluster composition is stable, all objects are reassigned to clusters and the centroids are redefined as the mean of that new cluster sample (Madhulatha, 2012). The silhouette score is defined by the average similarity of a point within its own cluster, $a(i)$, and the average dissimilarity between a point within a cluster and its closest neighboring cluster, $b(i)$ (Rousseeuw, 1987):

$$s_i = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

2.6 Bayesian Optimization

Bayesian optimization has a variety of applications in many different fields where automatic parameter-finding is needed; such as in robotics, information extraction, and automatic machine

learning (Shahriari et al., 2016). The purpose of this method is to maximize or minimize some blackbox objective function. The algorithm selects a point at which to observe the blackbox function based on an optimal acquisition function, which characterizes beliefs about an uncertain quantity (that of the objective function); a loss function to minimize is then required to assess the optimality of the result, which in this case is calculated from the residuals between two monkey datasets. The generic algorithm, as detailed in Shahriari et al. (2016) is:

Algorithm 1 Bayesian Optimization

1: **for** `n = 1, 2, ...` **do**
2: select new x_{n+1} by optimizing acquisition function α
3: $x_{n+1} = \text{argmax } \alpha(x; D_n)$
4: query objective function to obtain y_{n+t}
5: augment data $D_{n+1} = \{D_n, (x_{n+1}, y_{n+1})\}$
6: update statistical model
7: **end for**

This algorithm is implemented by the Spearmint package, as discussed in Section 3.5, and it can be used for numerous applications.

Chapter 3

Approach

3.1 Data Storage

The data was originally taken from PlasmoDB, an online repository of datasets related to *Plasmodium* experiments, from various labs. It is free for public use, and data taken in this experiment was collected by Joyner et al. Datasets are available as CSV files, and thus can be easily loaded with Python. Initially, the new files generated were easy to track and store as serialized files; however, even with only one user accessing the data, it became impractical to store these data and log different file-naming conventions. Therefore, data was migrated to MongoDB, a NoSQL document database that stores records in a JSON-like format; I chose this framework to be consistent with the Spearmint package described in 3.5. Mongo databases have collections, which contain records; when an experiment file is read, a new collection called `<experiment_name>` is created, and a record for each monkey and its data is stored as follows:

```
{
  "name": <monkey_name>,
  "raw_data": <BSON: raw data saved as Pandas DataFrame Object, with each datapoint as a row>,
  "norm_data": <BSON: normalized data saved as Pandas DataFrame Object, with each datapoint as a row>,
  "params_norm": <BSON: parameters fitted to normalized data saved as Pandas DataFrame Object, with each
     datapoint as a row>,
  "params_raw": <BSON: parameters fitted to raw data saved as Pandas DataFrame Object, with each datapoint
      as a row>,
  "days": <integer value of how many days the experiment ran for this monkey>,
}
```

Because various clinical parameters have large values, data was scaled appropriately (i.e. `data = log(data)`) if values were $>= 1$. Missing data was replaced with the average over all data for that parameter, as some values in the CSV files were "N/A". I used scaling and replacement of missing data as noted in Gomez (2010) for better results; initially I had replaced missing values with zero, but this greatly skewed Gaussian fits, since the peak-finding function as described in 3.2.0.1 would find those areas as false 'peaks'.

In the case of E04, the data entailed five monkeys, where each had 18 different clinical parameters stored. As further described in Section 3.4.1, because data was available only up to day 23 for one monkey as a result of terminal complications, analysis was performed across all 100 days for four monkeys (resulting in six pairs) as well as for only up to day 23 (resulting in 10 pairs). The raw and normalized data, as well as all Gaussian fits (see Section 3.2), have data for each clinical parameter as defined in Table B.1.

3.1.1 Normalization of Data

Normalization was used in this project at different stages of data analysis. As mentioned in 3.1, raw data was stored and normalized. While there are many different techniques to normalize data, in this project normalization was performed row-wise (i.e. with respect to that clinical parameter for that monkey), where the value for one day was divided by the sum of all values for that parameter. The data over all days was then a unit vector, which should have theoretically allowed for better comparison of different clinical parameters and different monkeys. Some results show that instead normalization reduced important magnitude-related variance that helped characterize the data. I also tried the `sklearn MinMaxScaler`, which restricts every value to a certain range (default (0, 1)) by using min-max normalization as in 2.3; this method returns slightly larger values than those from unit vector normalization over all 100 days, so I believed it could yield better results.

Moreover, I did not use normalized parameters from Gaussian fits for Bayesian Optimization, since "[c]omputers can get confused by very small or very large numbers, and round-off errors can result in misleading results." (Motulsky & Christopoulos, 2004); instead, I fit parameters to scaled raw data, and then normalized data after reconstruction over all days (using either unit vector normalization or min-max normalization). For example, if a clinical parameter has four Gaussian functions, these are fitted according to scaled raw data; the maximum value from all four Gaussians is then calculated for each day, and the resulting dataset is finally normalized.

3.2 Gaussian Fitting

The Gaussian function is defined as:

$$f(x|amp, \mu, \sigma) = amp * \exp \frac{-(x-\mu)^2}{2 * \sigma^2} \tag{3.1}$$

In order to better manipulate data, we fit multiple Gaussian functions to each clinical data-point, using the `sklearn curve_fit` function. Different functions can be used for curve fitting, which involve characterizing a better fit via a user-defined loss function (as in the `sklearn leastsq`) or a package-derived loss function (as in `sklearn curve_fit`). For example, `curve_fit` yielded better results than `leastsq`, as shown in Figure 3.1.

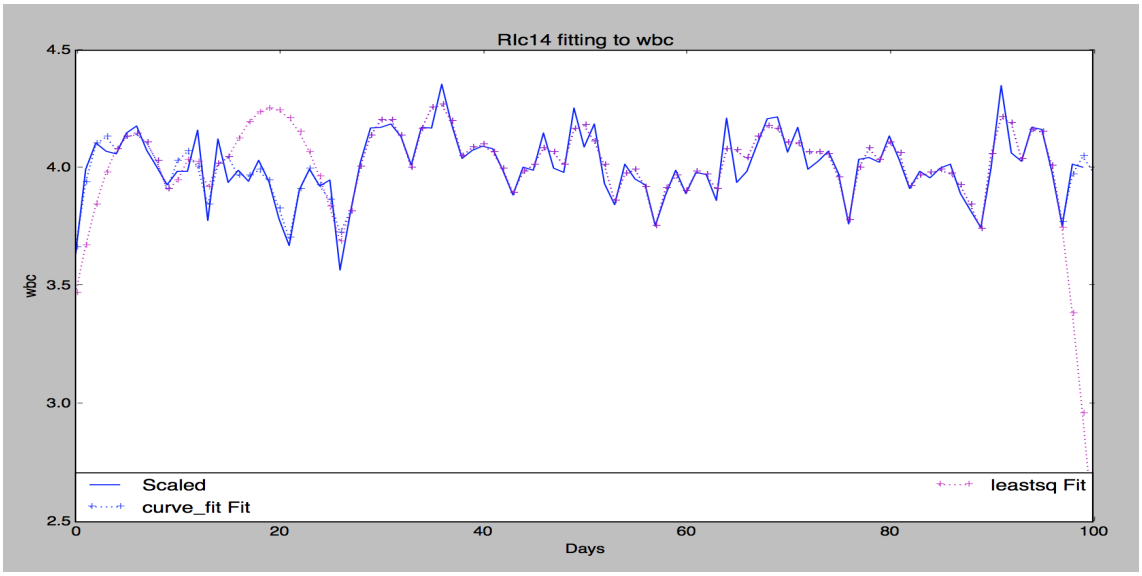


FIGURE 3.1: Comparing fits for an example monkey white blood cell count, using `curve_fit` and `leastsq`. As shown by the pink and blue curves, `curve_fit` provides better results.

The parameters to fit are *mean*, *standard deviation*, and *amplitude*, as denoted in equation 3.1. Finding an initial search space is very important for the `curve_fit` function, as otherwise it does not converge. Initial parameters were guessed as follows:

Parameter	Guess
Sigma	standard deviation normalized by maximal amplitude
Mean	location of the peak in that data range
Amplitude	magnitude of difference between the max and min points in the data

Although many of the data-points had very flat peaks, these were not reflected in the general calculation for standard deviation, since variance was low; thus, in order to construct an appropriate search space, the standard deviation was normalized by the maximal value in that data range such that flat peaks could be detected automatically:

$$standard\_dev = \frac{standard\_dev}{max(data)}$$

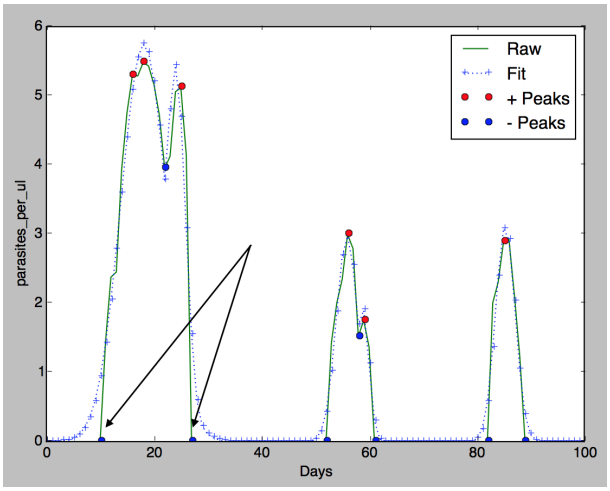
As an example, if all data-points within a fitting window are between 0 and 0.5, the standard deviation will be small; dividing by 0.5 in that case will double the standard deviation to help initialize that parameter closer to its optimal value.

The reasoning for this is also confirmed in (Motulsky & Christopoulos, 2004), as very large or very small y-values in data can cause errors regarding rounding or overflow. In addition, the location of the mean was predicted as the location of the peak; initially, the parameter search space was not bound, which caused overlapping of fitting windows. Consequently, during the later phases of data reconstruction over the entire range, the new peaks produced did not adhere to those in the raw data; to fix this discrepancy, the search space was bounded by the x-range for that data. For example, a section of (x, y) data may be from days 20-30; thus, the mean would start at whichever value corresponds to the peak y-value, and it would be bounded on

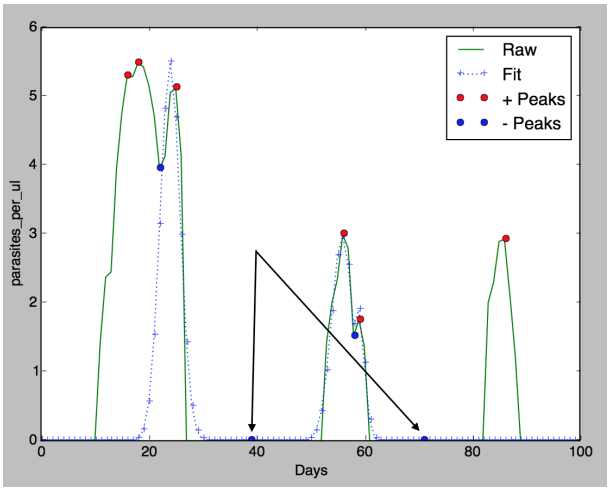
the interval [20, 30]. Each of these data sections is determined by the number of peaks found, as further described in 3.2.0.1.

3.2.0.1 Peak finding

Given normalized or raw data, sections of Gaussians were fitted based on maxima and minima. I derived a peak-finding algorithm to have a simplified, customized version. The `scipy` peak-finding algorithm requires previous knowledge of peak width, which is impossible in this project; `peakutils` from pypi is better since width is not necessary and peak threshold can be dictated, though it does not find the flat plateaux like algorithm 2. As seen in Figure 3.2, negative peaks in (b) are found in between Gaussians rather than at edges, as shown in (a). Peaks found at the edges of these Gaussians are crucial for finding optimal fitting windows, so that the `curve_fit` function can find parameters appropriately.



(a) Custom Peak Function



(b) Peakutils

FIGURE 3.2: Comparison of customized peak fitting and peakutils results; black arrows show examples of peaks found. The log-scaled count of parasites / uL is on the y-axis, while days of the experiment are on the x-axis.

The threshold used was 10% of the maximum value for each peak, and thus the peak algorithm as described below finds peaks by filtering out those that are less than threshold or that are false plateaux, while including true plateaux (e.g. not a flat line at 0). The 10% threshold allowed for Gaussians to be found, while not including very small values that could

Algorithm 2 Finding Peaks

```
1: procedure PEAKS(a, b)                    ▷ Find peaks greater than threshold, given data
2:     peaks ← []
3:     slopes ← numpy.diff(slopes)
4:     filteredSlopes ← filt(data, slopes, threshold)  ▷ removes areas of false plateaux (flat at
   0) and less than threshold
5:     for i in range(0, len(filteredSlopes)-1) do
6:         if (filteredSlopes[i] is not None)
7:     and ((filteredSlopes[i+1] < 0.0 and filteredSlopes[i] > 0.0)
8:     or (filteredSlopes[i+1] <= 0.0 and filteredSlopes[i] == 0.0)) then
9:             peaks.append((i+1))
10:        end if
11:    end for
12:    return peaks                                              ▷ Found peaks
13: end procedure
```

be considered noise. Each section was therefore determined by two negative peaks, as the peak-finding algorithm was run on the data (to find positive peaks) as well as the data with reversed signs (to find negative peaks). Using this algorithm, I could then automatically find appropriate sections on which to fit Gaussian functions.

An example of fitting is as shown in Figure 3.2(a); the peak-finding algorithm divides the x-range of days into appropriate sections on which to fit different Gaussian functions, with the parameters as dictated in Table 3.1. In order to reconstruct the data from the various Gaussian functions, the maximum value over all functions found is taken for each day; therefore, the bounds on the peak index parameter are crucial to reproducing the data correctly, as they restrict the proper window for that particular day (e.g. a value for day 25 should not be calculated from a Gaussian found in a window from [58, 61]).

TABLE 3.1: Parameters found corresponding to Figure 3.2(a)

X-range Values From Peaks	Gaussian Parameters Found (in parasites / uL)		
	amplitude	peakIndex	stdDev
10 - 22	5.75	18.11	4.25
22 - 27	5.44	23.96	1.92
52 - 58	2.96	55.90	2.00
58 - 61	1.92	58.83	1.13
82 - 89	3.12	85.32	1.82

3.3 Regression Modeling

`sklearn` has linear regression models available that each have different parameters and applications. At first, I used the generic Linear Regression model; however, I needed to resolve the problems of collinearity among different clinical parameters and of over-fitting, and thus I used the Ridge model because it implements *l2*-regularization (as described in 2.4). I chose to predict parasites per microliter, as it is a good indicator of disease severity; hence, the resulting coefficents would provide insight about what clinical parameters might be useful in understanding disease progression.

3.3.1 Combined Model

I also constructed a combined model to predict parasites per microliter based on multiple monkeys, using the `sklearn` Stochastic Gradient Descent Regressor (SGD). For this method, predictions from every monkey are first calculated separately for a given day. Next, these predictions are combined into a single output value based on certain weights assigned to each monkey, which are learned in the fitting process. Therefore, a final prediction is made that combines estimations from all monkeys, over all days. The SGD package is useful for combining models, as it provides *l2*-regularization and an error tolerance parameter. Stochastic Gradient Descent is the process by which optimal parameters are found via taking random walks, updating the error based on the new values. In the case of the SGD, this process is used for updating the loss function, while parameters are also penalized by regularization (*l1* or *l2*, or a combination of both- I chose *l2*).

3.3.2 'Phased' Regression

Because *P.cynomolgi* infections are characterized by relapses, I experimented with fitting regression models on phases (specified x-ranges) to assess if the resulting predictions for other monkeys were more accurate. In this way, if phases could be found automatically and used to predict other monkeys, the accuracy of those fits could be used as another metric for similarity between monkeys of different phenotypes (to indicate level of severity). This process entails breaking the experiment period (100 days) into smaller windows and fitting a regression model to that small window, or 'phase'; then, this more specific model can be used to predict a different monkey in the same 'phase', and the error in the prediction can be used as a metric of similarity where lower error signifies a more similar monkey. To find phases automatically, I continued using the

same peak-finding function (see Algorithm 2), assuming that phases could be defined by the same windows I used for Gaussian fitting.

3.4 Clustering

The basic idea of clustering is to glean information from unlabeled data, and there are many different clustering algorithms available. In this study, I used Ward Agglomerative, Gaussian Mixture, Spectral, and k-means clustering methods. I chose these methods experimentally, as `sklearn` has existing modules available; ultimately I used k-means (Section 2.5) for analysis, as it had the most consistent silhouette score (see Chapter 4). This metric considers both inter- and intra-cluster similarity (Section 2.5), and so it was appropriate for evaluation.

In order to cluster data, however, vectors must be constructed to create a spatial representation of the data. Vector visualization in space can be shown by projecting the high-dimensional data into only two dimensions, as with Principal Component Analysis in Figure 3.3. As explained further, I used a few different methods to quantify clinical parameters in vector space.

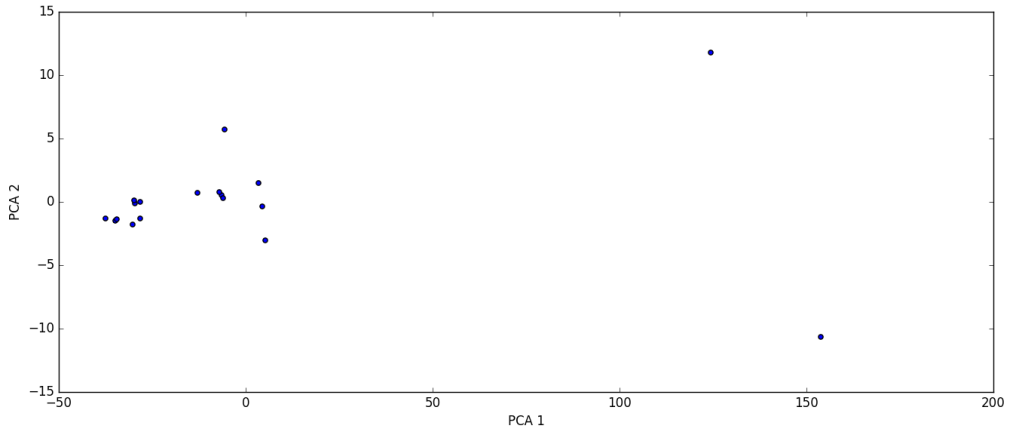


FIGURE 3.3: Example of projecting the 18D residual matrix using PCA (matrix dimensions explained in Section 3.4.1).

3.4.1 Residual Matrices

The first way to represent clinical data in vector space is to create a residual matrix, which characterizes variance among monkeys with respect to each parameter (allowing parameters that vary similarly to cluster together). In the matrix, constructed pairwise for all monkeys, each row is a clinical parameter and each column is a pair of monkeys. Residuals are calculated as follows, where $y1$ and $y2$ are two sets of y-values and n is the number of x-values in common (e.g. for the monkey that died early, the residual is calculated only up to that day; otherwise, n = 100):

$$residual(y1, y2) = \sqrt{(y1_0 - y2_0)^2 + (y1_1 - y2_1)^2 + ... + (y1_n - y2_n)^2} \tag{3.2}$$

Because the initial experiment includes a monkey that died early, the matrix in Table 3.2 has only the other four because residuals are calculated over 100 days; a different matrix was

created for monkeys up to the twenty-third day. The final matrices for experiment E04 are either 18 x 6 or 18 x 10 (four monkeys or all five monkeys) dimensions.

TABLE 3.2: Residuals found corresponding to Figure 3.3, parameters as defined in Table B.1

	RIc14_RFa14	RIc14_RSb14	RIc14_RMe14	RFa14_RSb14	RFa14_RMe14	RSb14_RMe14
gran	10.39	15.61	16.79	13.60	15.98	10.26
hct	4.25	3.82	5.64	3.33	3.00	4.52
hgb	4.40	3.87	5.58	3.40	3.18	4.09
lymph	8.57	11.93	13.71	12.02	14.03	6.20
mch	1.36	2.14	1.13	3.30	1.14	2.22
mchc	0.86	0.43	0.57	0.82	1.23	0.67
mcv	1.60	2.10	0.86	3.68	2.01	1.71
mono	17.09	14.10	13.80	14.69	15.31	9.21
mpv	3.05	4.96	3.90	7.14	5.88	2.69
parasitemia_perc	18.51	11.56	16.97	20.36	23.18	13.61
parasites_per_ul	75.19	57.78	55.48	67.88	74.42	69.48
plt	11.16	18.69	13.77	12.93	9.84	15.60
rbc	5.50	2.77	5.76	5.37	3.20	5.27
rdw	3.02	3.91	3.73	3.12	3.42	5.48
reticulocytes_num	15.55	19.45	14.16	15.99	19.76	24.30
reticulocytes_per_ul	17.68	18.37	15.08	13.46	20.78	21.87
reticulocytes_perc	67.95	61.84	65.90	84.11	96.76	90.65
wbc	11.80	13.30	21.75	7.61	14.94	14.81

3.5 Bayesian Optimization

I used Bayesian Optimization, as described in Section 2.6, to find the optimal parameters for shifting monkey data such that the residual between two given monkeys is minimized. As previously described, I chose to characterize clinical data with residuals between monkeys to

show which data-points varied similarly, especially regarding monkeys of the same phenotype; therefore, if the pattern of residuals is similar between two data-points, then those clinical data may contain the same information with respect to determining severity of the malaria phenotype. However, because biological variation is expected even among monkeys that have similar symptoms and final outcomes, just as in humans, shifting the data pairwise among monkeys can help to minimize these differences and better select important clinical parameters (i.e. for one pair of monkeys, the data for each monkey in that pair is shifted to minimize the residual between the two monkeys).

The reason for using this method in general is that it can automatically derive insight from a very complex function. In this project, it would be impossible to find optimal shifting parameters manually, as the shifting window has seven values (from [-3, 3], window chosen experimentally) and could have even twenty Gaussian functions for each monkey (yielding 7^{20} possibilities for one monkey x 7^{20} possibilities for the other); with this number of possibilities, it would also be incredibly expensive to use brute-force methods. Bayesian optimization is therefore a natural choice to guess information about the objective function, yielding shifting parameters appropriate for minimizing the residuals. Rather than re-implement this complex method, I used a package called Spearmint.

These new regression models were run with respect to pairs of monkeys (e.g. monkey1 is shifted with respect to monkey2 and vice versa, and their models are refitted). In addition, the combined models as described in 3.3.1 were re-run with respect to one monkey; as an example, the regression model for monkey1 was kept constant, while the models for all other monkeys were shifted with respect to monkey1 (using the shifted parameters found from Spearmint) to culminate in a combined prediction model tailored to that monkey.

3.5.1 Spearmint Package

The Spearmint package (Snoek et al., 2012) requires a loss function on which to gauge the optimality of parameters, and it was determined by the residual between two monkeys; this residual was calculated by:

```
residual = get_max_gaussians(get_shifted_gaussians(monkey1, shift1),
get_shifted_gaussians(monkey2, shift2))
```

The Gaussians fitted previously on the data are thus shifted by the parameters guessed in the objective function, and then the residual is calculated by reconstructing the entire set of data for all days (calculate the maximum value over all gaussian parameters for that day), given the new parameters. Spearmint subsequently guesses the next set of shift parameters that may reduce the residual in the next run, allowing the statistical model to be updated at each experiment. This package uses MongoDB to save experiments, and so I could access the results later when the experiments finished. The more shifting parameters there are (i.e. more Gaussian functions fitted), the longer this optimization takes; hence, I modified the code to stop after thirty runs. I was able to use this package only after fixing many issues regarding MongoDB, configuration files, and 'broken' experiments (term the package uses for experiments that did not run), as it is very difficult to debug (see Section 5.3).

After finalizing shifting parameters for each pair of monkeys, I could then re-run all analyses performed previously with non-shifted data to determine how minimizing residuals between monkeys might improve machine understanding of the data.

3.5.2 Sign-Match Matrices

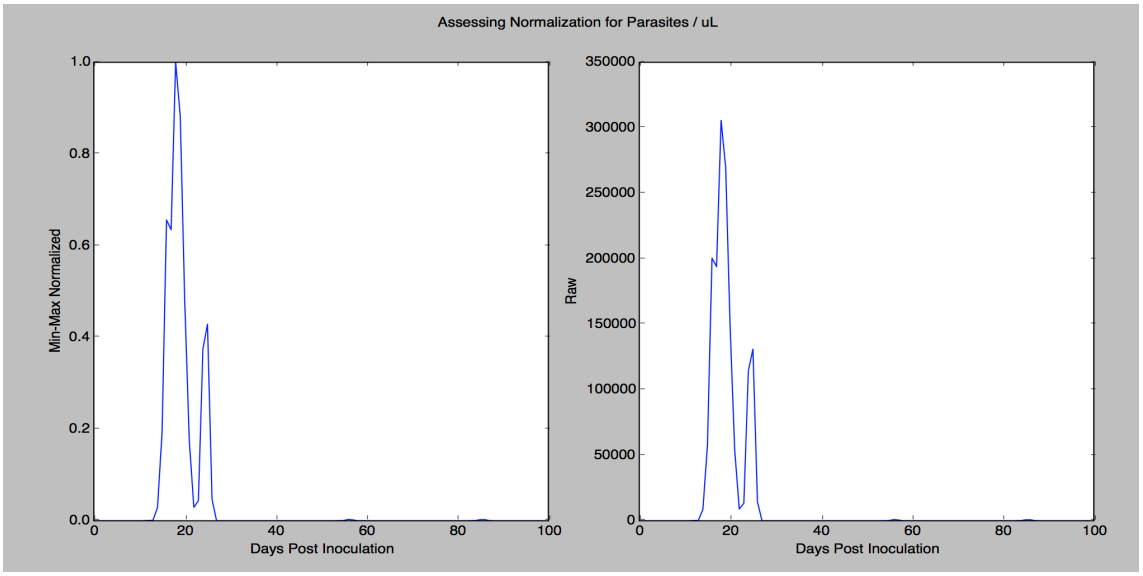
As mentioned previously, each clinical parameter must have some kind of metric for representation in vector space. Residual matrices are one way, but we decided to record matching signs before and after shifting to quantify how much shifting helped clarify similarities and differences among data-points. Regression coefficients can show trends with negative or positive values, with the coefficient magnitude as the weight of that data-point; recording which signs match can then show any similar trends among data-points. Thus, determining the usefulness of shifting can be done not only by comparing the same residual matrices before and after shifting, but also by looking at whether or not signs match between monkeys. This helps both to represent differences between monkeys of varying phenotypes and to reveal similarities between data-points having similar trends.

Chapter 4

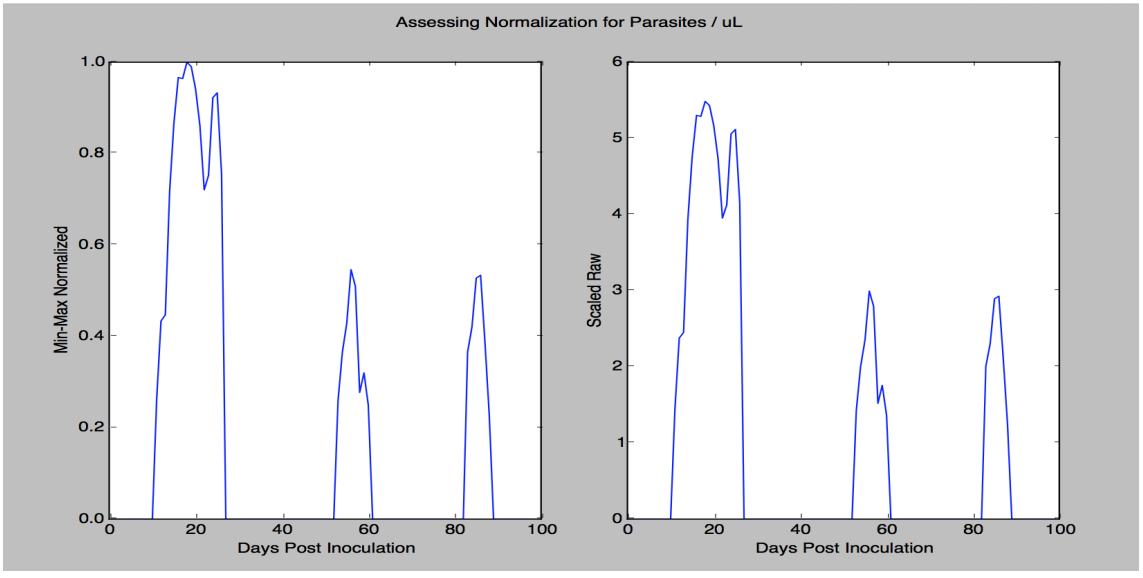
Results

4.1 Gaussian Fitting and Normalization

The resulting fits to the scaled data worked quite well, as shown by the example mean squared error in figure Figure 4.2 and Figure 4.3. There is a stark difference to note in the normalized data that was cautioned in (Motulsky & Christopoulos, 2004); because the values for parasites per microliter are so large, the middle and end parasite counts are reduced to extremely small values, leaving only the first phase with ample data, as shown explicitly in Figure 4.1. Therefore, normalization must be applied to scaled data, in order to be meaningful. Regarding the other two example data-points, lymphocytes and platelets, the integrity of the curves is kept even after normalization to raw data.



(a) Raw Data



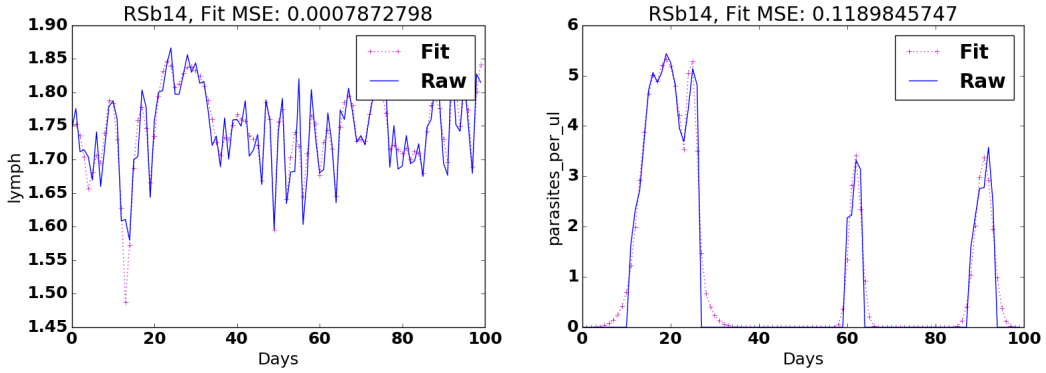
(b) Scaled Data

FIGURE 4.1: Comparing min-max normalized data to raw data (a) and scaled data (b), for an example monkey; as seen without scaling, the parasite count in the next two phases is completely diminished, which could be removing important biological variation that helps to distinguish different phenotypes. However, scaling helps to keep the integrity of phases even after normalization.

Parameters fitted to scaled raw data were used for shifting in Bayesian Optimization rather

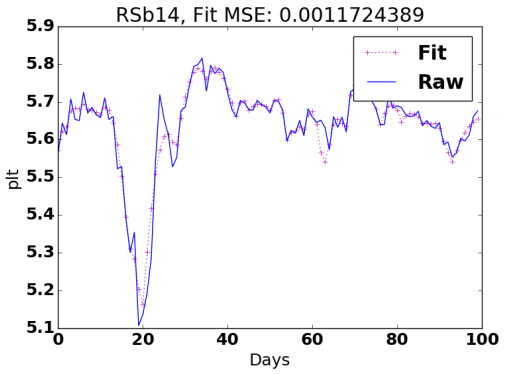
than those fitted to normalized data or normalized scaled data; the acquisition function in Spearmint can better update parameters based on a larger loss function, which is characterized by larger initial values (i.e. not normalized). Normalization is thus useful later in clustering after all values have been shifted accordingly, in order to equalize clinical parameters in vector space. As discussed in Section 4.3.1, normalization was not useful in creating some of the residual matrices, while it was helpful for some clustering results.

Figure 4.3 shows a proof of concept, however, in that fitting to small data values as a result of normalization is possible with the Gaussian fitting methodology explained in Section 3.2, especially in using the initial value for standard deviation normalized by the maximum y-value. These results confirm that the first step in our data analysis framework, fitting Gaussian functions to biological data, is viable.



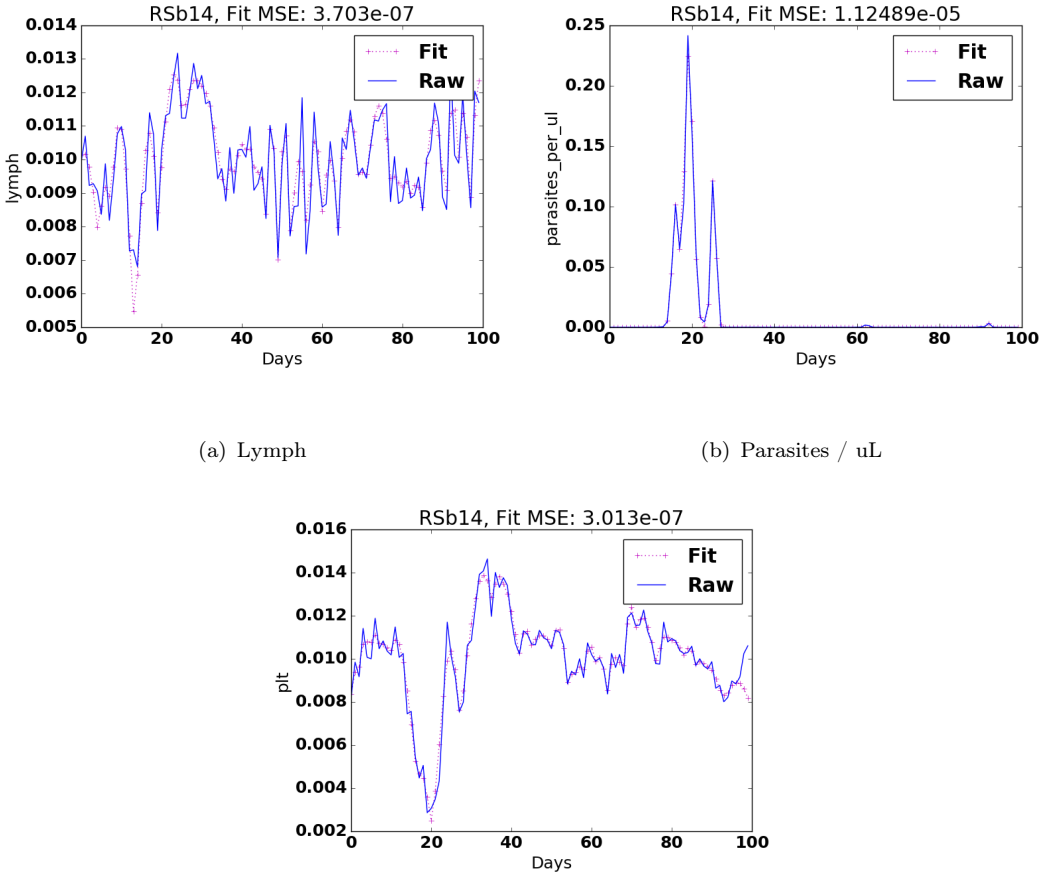
(a) Lymph

(b) Parasites / uL



(c) Platelets

FIGURE 4.2: Example output of different Gaussian fits for monkey RSb14, fitted to scaled data; MSE (mean squared error) is as shown on each subfigure.



(a) Lymph

(b) Parasites / uL

(c) Platelets

FIGURE 4.3: Example output of different Gaussian fits for monkey RSb14, fitted to normalized data; MSE (mean squared error) is as shown on each subfigure.

4.2 Regression Modeling

The normalized mean-squared error ranges from 0.0054 to 0.0081 per day for the regression models to predict the number of parasites per microliter, as seen in Table 4.1 . The lowest error is in predicting RFv13 and RMe14, which are the lethal and very severe phenotypes, respectively. This observation is interesting because it suggests, in accordance with Table 4.4 and Table 4.5, that the more severe phenotypes have more predictive, or extreme, features. Also as seen in

Table 4.1, the more severe or lethal phenotypes have a larger average magnitude of coefficients as compared to the non-severe monkeys. These analyses show that there are possible markers for the severe vs. non-severe phenotypes.

TABLE 4.1: Mean squared error for the monkey graphs, normalized against the number of days.

Monkey	Normalized MSE	Avg coeffs	Avg coeffs (abs value)
RFa14 (sev)	0.0081	-0.154	4.63
RSb14 (non-sev)	0.0071	-0.514	2.18
RIc14 (non-sev)	0.0099	-0.619	3.26
RMe14 (very sev)	0.0066	-2.56	5.65
RFv13 (lethal)	0.0054	-0.226	3.84

Moreover, looking at Table 4.2, the highlighted features represent notable differences (whether in sign, magnitude, or proximity to zero) between the monkey of the lethal phenotype and the others; in order to compare coefficients from different monkeys better with values normalized to the range [-1,1], the *tanh* function was applied to the the results as shown in Table 4.3. *tanh* is commonly used as an activation function in machine learning for analyzing the differences in output for neural networks; therefore, I chose to use this function to normalize coefficients while keeping the significance of sign differences. Even as the magnitude of the platelets coefficient in Table 4.2 for RFv13 was larger in comparison to the other monkeys, it was more similar to the other monkeys in Table 4.3. *tanh* can therefore help to highlight differences that are more significant among monkeys, normalizing those that are due to noise or biological variation unrelated to disease phenotype (rather than minimized variation that diminishes important distinguishing factors). The two features that seem the most similar between the two non-severe monkeys are lymphocytes and reticulocytes per microliter; Joyner et al. discuss the possible importance of a modest increase in reticulocyte count between days 10-15 in the monkeys that survived,

in contrast to RFv13 (Joyner et al., 2016). The similarity found between the two non-severe monkeys is reflective of this finding, confirming that some insight can be found automatically. In addition, hemoglobin is stated to be negatively correlated in the paper; after applying *tanh*, the two non-severe monkeys have hemoglobin as positively correlated, while the other monkeys show a negative correlation. As a result, hemoglobin could also be an indicator for differences between severe and non-severe phenotypes. After application of the *tanh* function as well, red blood cell distribution width, rdw, is the largest positive value for the lethal monkey; greater variation in this parameter signifies illness, and thus the coefficients are reflective of this biological implication. Overall, regression showed interesting results and was rather accurate for predicting parasites per microliter, as seen in Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8 ("Gold" means actual data, versus fitted data).

TABLE 4.2: Weights for all features: regression models fit with respect to that monkey itself.

	Feature	RFa14	RSb14	RIc14	RMe14	RFv13
0	gran	0.83	1.21	3.69	3.61	-1.52
1	hct	6.23	-3.30	-3.51	-4.75	2.04
2	hgb	-4.50	2.24	4.48	-4.23	-1.26
3	lymph	-1.42	-0.41	-0.49	1.96	-5.67
4	mch	-3.45	-1.28	4.27	-2.58	0.63
5	mchc	18.42	5.57	4.81	2.22	2.15
6	mcv	-4.34	-3.66	-3.93	-6.97	-0.39
7	mono	1.11	0.56	1.88	2.24	3.65
8	mpv	-2.01	-0.14	-5.31	-15.48	-3.61
9	plt	-5.00	-9.34	-6.83	-2.75	-15.63
10	rbc	4.93	1.18	-2.98	12.39	2.28
11	rdw	-8.82	0.16	-3.35	-21.60	2.99
12	# ret	3.94	1.75	1.33	-1.89	1.45
13	ret / uL	-3.35	0.67	0.66	1.96	8.70
14	ret %	0.34	-0.17	0.01	0.31	-4.42
15	wbc	-5.38	-3.27	-4.63	-5.42	4.99

TABLE 4.3: Tanh applied to weights for all features: regression models fit with respect to that monkey itself.

	Feature	RFa14	RSb14	RIc14	RMe14	RFv13
0	gran	0.680476	0.836679	0.998754	0.998537	-0.908698
1	hct	0.999992	-0.997283	-0.998214	-0.999850	0.966747
2	hgb	-0.999753	0.977587	0.999743	-0.999577	-0.851064
3	lymph	-0.889599	-0.388473	-0.454216	0.961090	-0.999976
4	mch	-0.997986	-0.856485	0.999609	-0.988582	0.558052
5	mchc	1.000000	0.999971	0.999867	0.976683	0.973226
6	mcv	-0.999660	-0.998677	-0.999229	-0.999998	-0.371360
7	mono	0.804062	0.507977	0.954492	0.977587	0.998650
8	mpv	-0.964727	-0.139092	-0.999951	-1.000000	-0.998537
9	plt	-0.999909	-1.000000	-0.999998	-0.991860	-1.000000
10	rbc	0.999896	0.827452	-0.994853	1.000000	0.979293
11	rdw	-1.000000	0.158649	-0.997541	-1.000000	0.994955
12	# ret	0.999244	0.941376	0.869249	-0.955373	0.895693
13	ret / uL	-0.997541	0.584980	0.578363	0.961090	1.000000
14	ret %	0.327477	-0.168381	0.010000	0.300437	-0.999710
15	wbc	-0.999958	-0.997115	-0.999810	-0.999961	0.999907

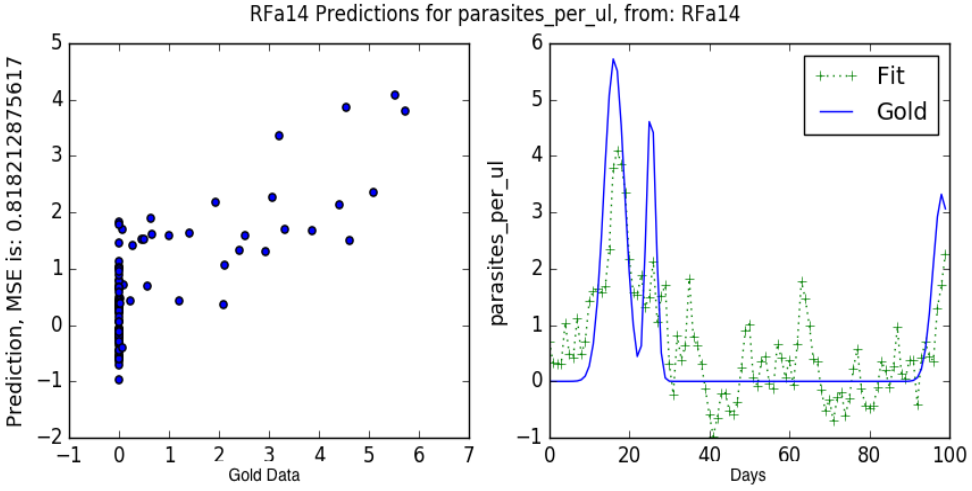


FIGURE 4.4: Regression model fitting for the monkey (RFa14) to itself and from a combined model of all monkeys, predicting parasites/ uL.

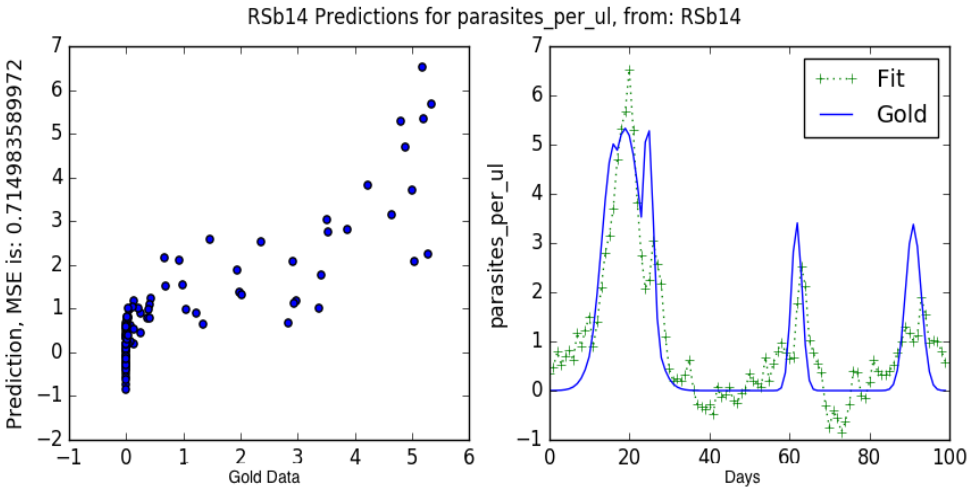


FIGURE 4.5: Regression model fitting for the monkey (RSb14) to itself and from a combined model of all monkeys, predicting parasites/ uL.

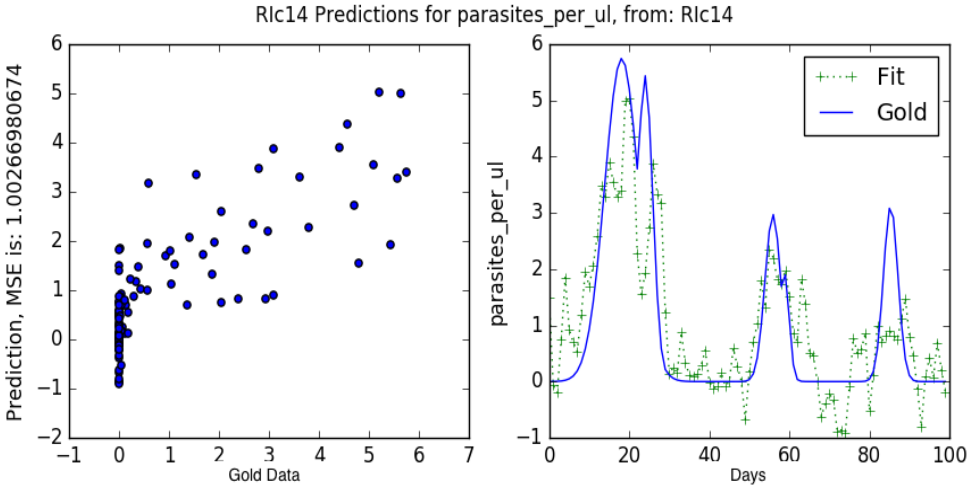


FIGURE 4.6: Regression model fitting for the monkey (RIc14) to itself and from a combined model of all monkeys, predicting parasites/ uL.

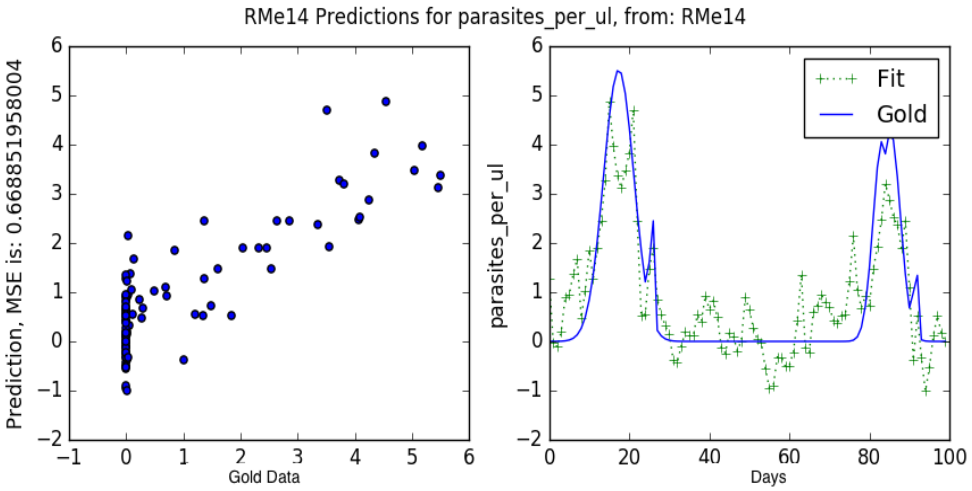


FIGURE 4.7: Regression model fitting for the monkey (RMe14) to itself and from a combined model of all monkeys, predicting parasites/ uL.

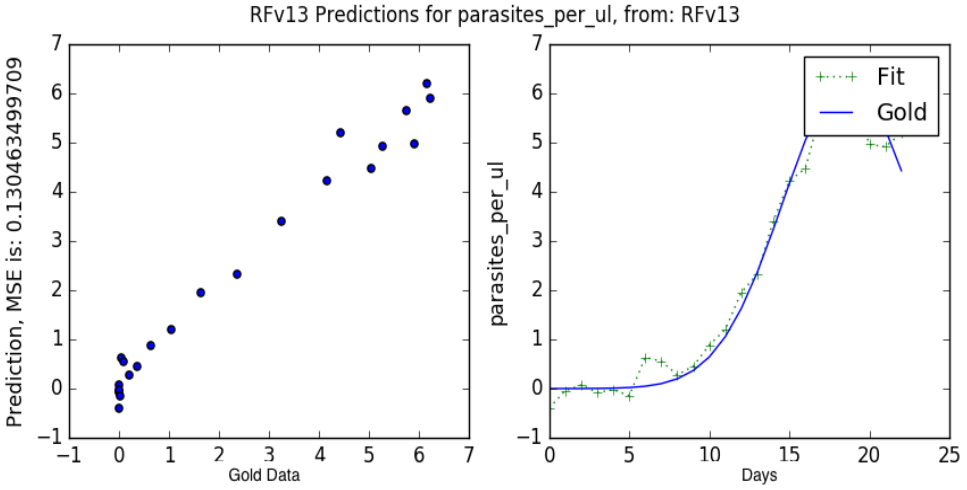


FIGURE 4.8: Regression model fitting for the monkey (RFv13) to itself and from a combined model of all monkeys, predicting parasites/ uL.

4.2.1 Combined Model with Bayesian Optimization Shifts

As seen in the tables below (Table 4.4, Table 4.5), the Ridge regression models were used in combination to predict the number of parasites per microliter in each monkey; this was done over all days of the experiment and for only up to day 23 (to include the monkey of the lethal phenotype). It is interesting to note that over only 23 days, the monkeys with non-severe phenotypes (RIc14 and RSb14) have the lowest coefficients. With the combined model, each monkey predicts daily values for the monkey of interest, and these values are then combined via the weights specified; therefore, since the non-severe monkeys have the lowest weight, the models show that monkeys with more severe phenotypes are more predictive of parasite count, particularly in the first phase of the infection.

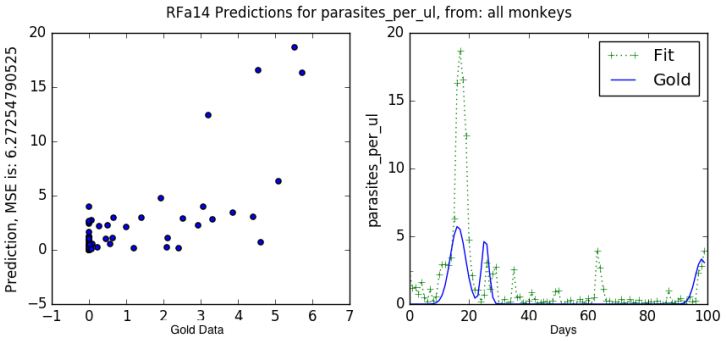
TABLE 4.4: Weights for all monkey models in predicting parasites / uL, over all days (exclude RFv13).

	Monkey	Coefficient
0	RIc14	0.05
2	RSb14	0.23
3	RMe14	0.20
4	RFa14	0.54

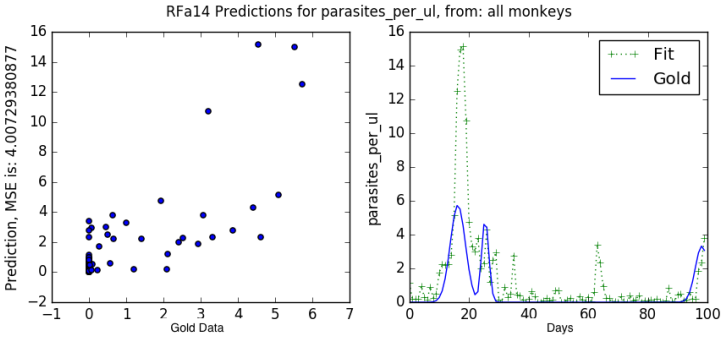
TABLE 4.5: Weights for all monkey models in predicting parasites / uL, until day 23 (include RFv13); distinctly low weights are highlighted.

	Monkey	Coefficient
0	RIc14	0.04
1	RFv13	0.13
2	RSb14	0.03
3	RMe14	0.25
4	RFa14	0.40

As seen in Figure 4.9, the MSE decreases after shifting with respect to RFa14, showing that the regression model for the combined monkeys is improved after Bayesian optimization.



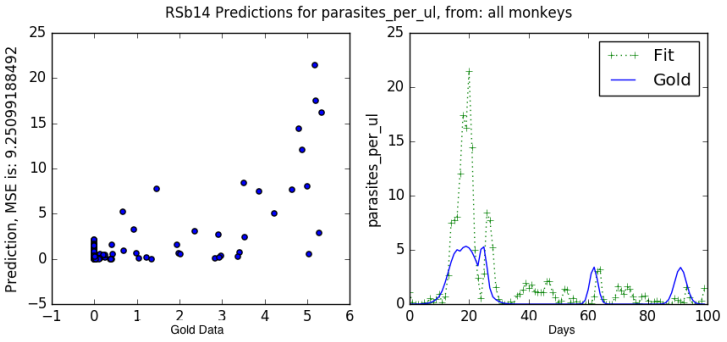
(a) Combined Prediction



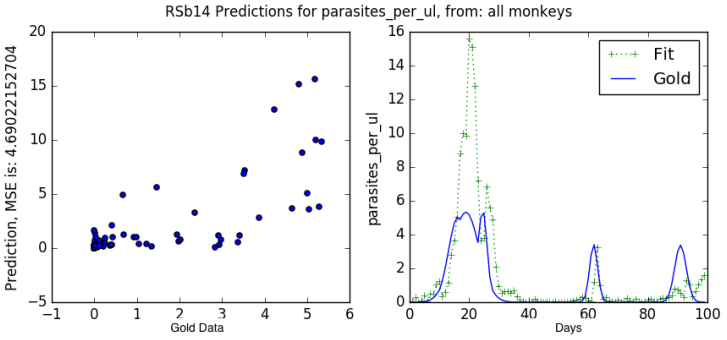
(b) Combined Prediction, Post Shifting

FIGURE 4.9: **RFa14**: Comparing combined regression models over all monkeys (exclude RFv13), with and without Bayesian Optimization shifts, predicting parasites/ uL.

This same trend is confirmed in Figure 4.10, reducing the MSE from 9.25 to 4.69, and especially so in Figure 4.11 and Figure 4.12 with reductions from 44.5 to 4.63 and 88.7 to 4.06, respectively.

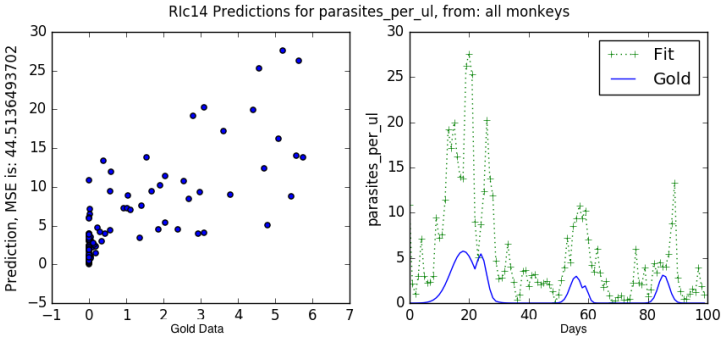


(a) Combined Prediction

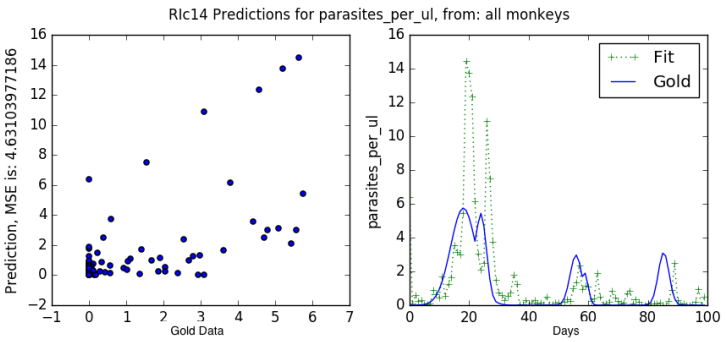


(b) Combined Prediction, Post Shifting

FIGURE 4.10: **RSb14**: Comparing combined regression models over all monkeys (exclude RFv13), with and without Bayesian Optimization shifts, predicting parasites/ uL.

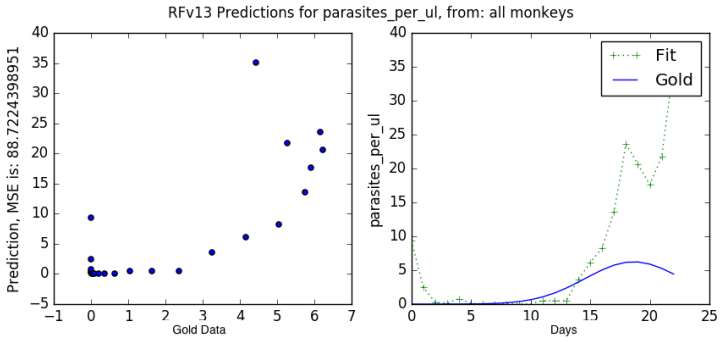


(a) Combined Prediction

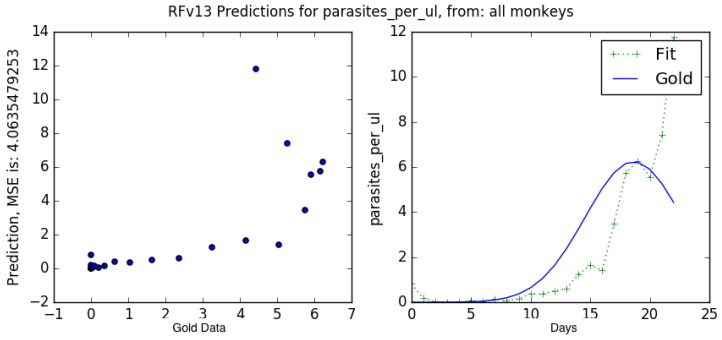


(b) Combined Prediction, Post Shifting

FIGURE 4.11: **RIc14**: Comparing combined regression models over all monkeys (exclude RFv13), with and without Bayesian Optimization shifts, predicting parasites/ uL.



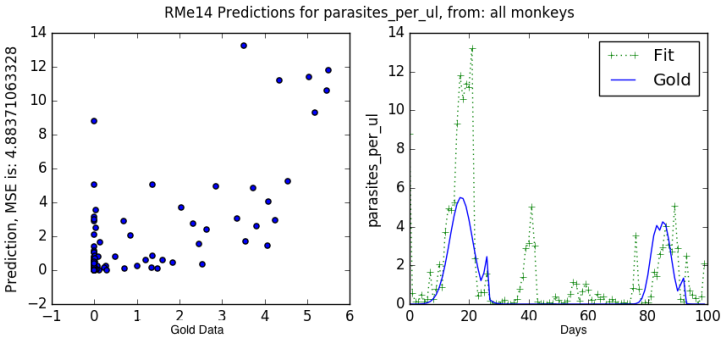
(a) Combined Prediction



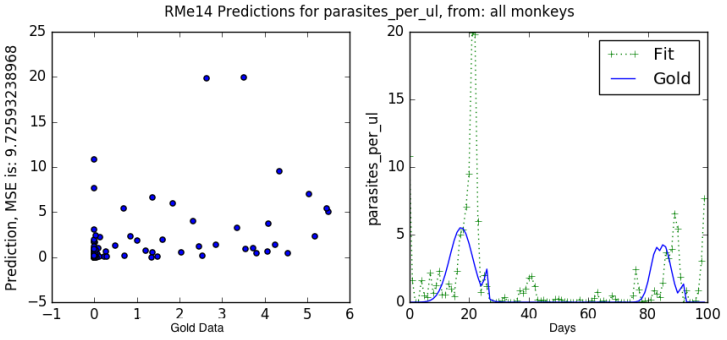
(b) Combined Prediction, Post Shifting

FIGURE 4.12: **RFv13**: Comparing combined regression models over all monkeys, with and without Bayesian Optimization shifts, predicting parasites/ uL.

Lastly, regarding RMe14 in Figure 4.13, while the MSE actually increases, the trend is more true to the actual data, since the middle peak predicted around day 40 is correctly diminished after shifting. These results confirm that using Bayesian optimization to find optimal shifting parameters, reducing the overall residual, helps create better models for analysis. After this verification, I could then continue on to further analysis with combined shifted models to determine if all monkeys, shifted and subsequently combined, could create a more accurate predictive model.



(a) Combined Prediction



(b) Combined Prediction, Post Shifting

FIGURE 4.13: **RSb14**: Comparing combined regression models over all monkeys (exclude RFv13), with and without Bayesian Optimization shifts, predicting parasites/ uL.

Furthermore, the two non-severe monkeys are best predicted by RSb14 and RFa14, one non-severe monkey and one severe monkey respectively, as shown in Table 4.6. In contrast, the two severe monkeys, RFa14 and RMe14, are best predicted by RFa14, with very similar weights across all monkeys. RFa14, of a severe phenotype, has the largest average coefficient calculated both using all coefficients and excluding its own coefficient. This result confirms again that phenotypes of greater severity may have more significance in predicting the number of parasites at a given time; because the parasite count also signifies the severity of the disease, it is logical that the monkeys of more severe phenotypes have greater weight in predicting infection. The reason for a low weight in RMe14, exhibiting a very severe phenotype, could be a result of its receiving a blood transfusion, which could have changed the normal, possibly canonical course

of biological fluctuations in a severe infection.

TABLE 4.6: Shifted with respect the monkey given: Weights for all monkey models in predicting parasites / uL, over all days (exclude RFv13) and for up to day 23 (include RFv13).

	Target monkey (shifted with respect to this monkey)						
Fitted Monkey	RFa14	RSb14	RIc14	RMe14	Rfv13	Avg Coeff	Avg (exclude self)
RIc14	0.12	-0.32	-0.01	0.12	0.16	0.014	0.020
RSb14	0.10	0.70	0.31	0.15	0.24	0.30	0.20
RMe14	0.01	-0.06	0.12	0.06	-0.17	-0.0080	-0.025
RFa14	0.74	0.49	0.70	0.78	0.23	0.59	0.55
RFv13					0.41	0.41	

4.2.2 'Phased' Regression

Using phased regression, as explained in Section 3.3.2, signifies fitting the regression model within a certain time-point and using that model to predict the corresponding time-point in another monkey, still using the parameters fitted to scaled raw data. The results show that this method does not yield better results for predicting other monkeys or predicting the same monkey itself. The mean squared error (MSE) for RFa14, over the entire 100 days, is 0.818, as shown in Figure 4.4. In comparison, in Figure 4.14, the MSE is 1.35 for only the first phase from days 0-18, while over the entire 100 days the total MSE is 7.528. The same trend is seen for all monkeys in Table 4.7, which gives MSE values for every phase. The two highlighted values are the largest MSE in the entire table, which result from pairwise shifting between the two non-severe monkeys. This trend reflects results from residual matrices for parasites per microliter, pre-shifting (Figure 4.25), as the residuals between the two non-severe monkeys were large before shifting. Therefore, the phased regression does not do well in predicting general phases because pre-shifting, the monkey phases can be different and therefore result in high-error predictions.

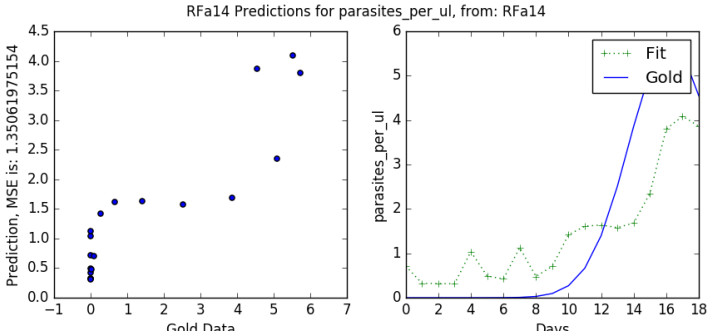


FIGURE 4.14: Regression model fitting for the monkey (RFa14) to itself for that phase, predicting parasites/ uL.

TABLE 4.7: Mean squared error for predicting the given monkey, based on a model fitted to a certain phase. The numbers in parentheses signify the beginning and end range of the phase; the monkeys at the top in columns are those with the regression model, while those in rows are predicted based on the monkey at the top.

	RSb14					
Monkey	(0, 17)	(17, 22)	(22, 27)	(52, 82)	(82, 99)	**sum**
RIc14	0.849	2.228	5.908	0.481	1.008	**10.474**
RFv13	6.737	16.629	0.011			**23.377**
RSb14	4.104	4.379	13.111	5.636	7.058	**34.288**
RMe14	2.338	4.764	3.405	3.494	6.992	**20.933**
RFa14	1.416	0.519	7.408	0.75	1.661	**11.754**

	RFv13	RSb14					
Monkey	(0, 22)	(0, 17)	(17, 23)	(23, 27)	(59, 87)	(87, 99)	**sum**
RIc14	7.373	7.991	2.858	4.232	6.422	8.314	**29.817**
RFv13	0.11	1.286	4.69				**5.976**
RSb14	5.214	0.682	0.393	4.697	0.526	1.322	**7.62**
RMe14	26.52	3.459	0.258	1.025	1.931	1.408	**8.081**
RFa14	10.832	2.632	2.895	8.427	2.931	2.061	**18.946**

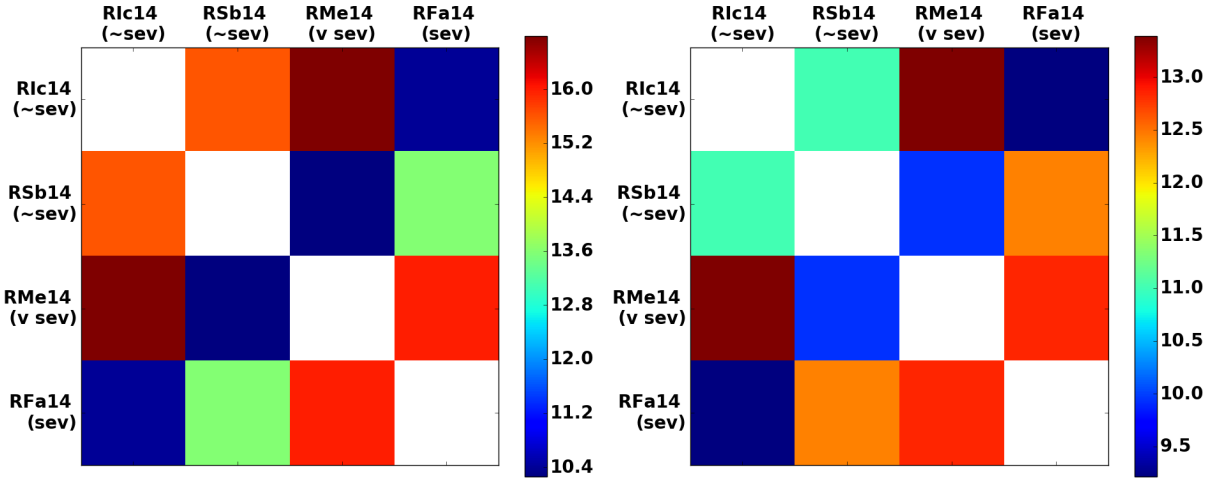
	RMe14					RFa14			
Monkey	(0, 22)	(24, 79)	(79, 84)	(84, 99)	**sum**	(0, 19)	(23, 27)	(94, 99)	**sum**
RIc14	3.453	2.095	1.5	0.99	**8.038**	2.741	3.463	1.747	**7.951**
RFv13	5.855					4.088			
RSb14	0.945	1.225	0.202	0.98	**3.352**	1.248	9.36	0.358	**10.966**
RMe14	1.131	0.395	1.385	0.768	**3.679**	2.476	0.591	0.843	**3.91**
RFa14	3.701	1.176	0.51	0.777	**6.164**	1.351	4.417	1.76	**7.528**

4.3 Clustering

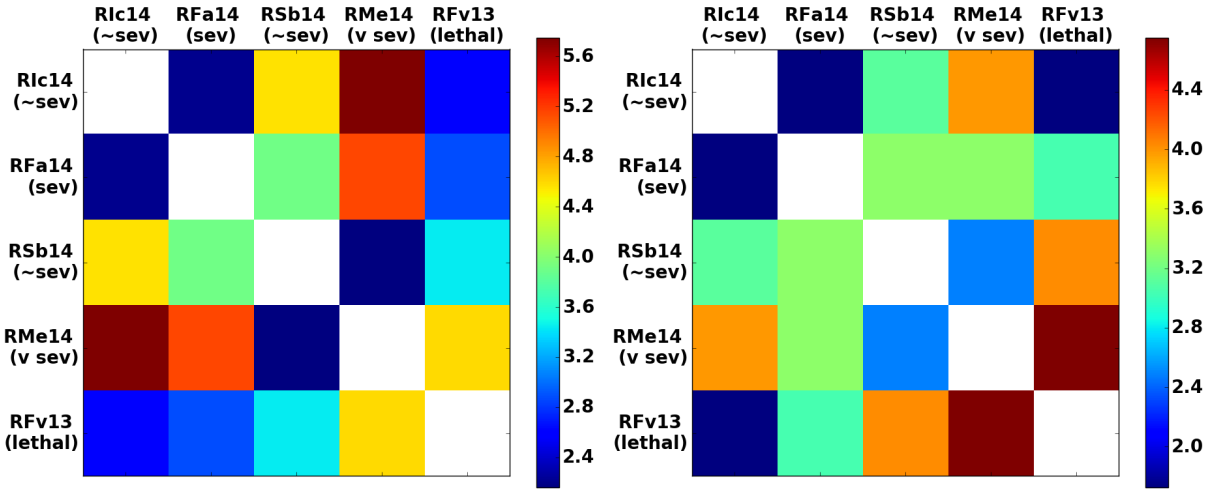
4.3.1 Residual Matrices

Residual matrices were constructed as described in Section 3.4.1, in order to quantify similarity among the monkeys. Moreover, residuals were minimized using Bayesian optimization to see which clinical parameters might be more indicative of disease characterization and severity. Using min-max normalization from `sklearn` yielded some helpful results, but others did not improve; all normalized matrices are included in the Appendix, in Section A.1.1, as only the final clustering and PCA results are explicitly discussed. Even as some normalized matrices did not 'improve' regarding grouping non-severe monkeys and severe monkeys together, however, normalization was still interesting clinical parameters were clustered in different ways, discussed further in Figure 4.35.

As seen in Figure 4.15 for granulocytes over all days, while the residual was reduced over all pairs of monkeys, it decreased the most among the non-severe monkeys, from about 16 to 11; this shows that shifting can help normalize the differences between monkeys to understand better the importance of a certain parameter. For only up to day 23, shifting shows that the monkeys were similar in general for number of granulocytes in the first phase of the disease, whereas differences are more pronounced across all days of the experiment in (a). Regarding Figure 4.16, it seems that while there is not a stark trend for the non-severe monkeys, the number of red blood cells per total blood volume is more similar among severe monkeys as compared to between non-severe monkeys; again, this trend is more important over the course of the entire experiment, rather than only for the first 23 days as shown in (b).

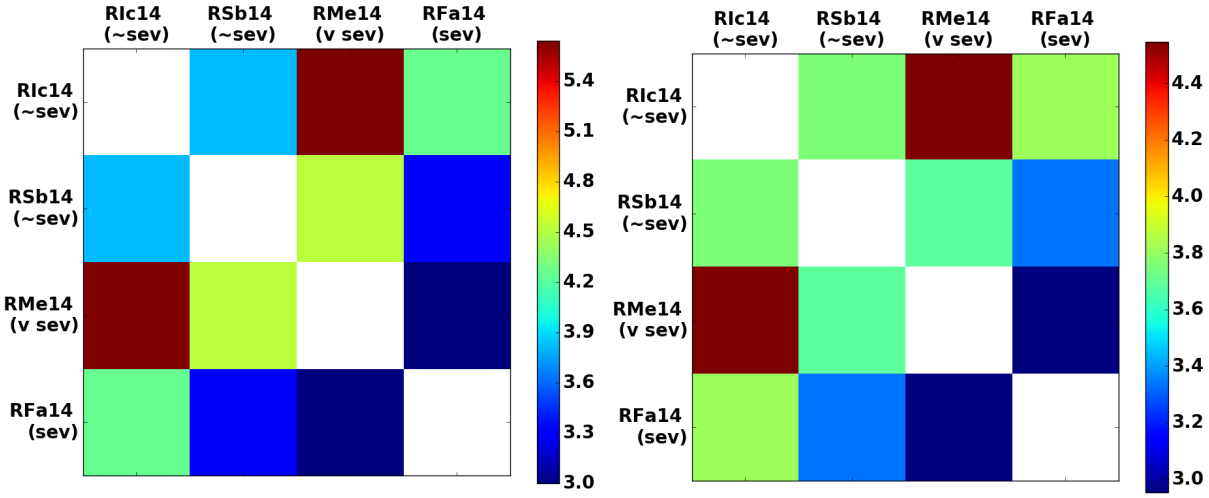


(a) Over all days, post-shifting on the right

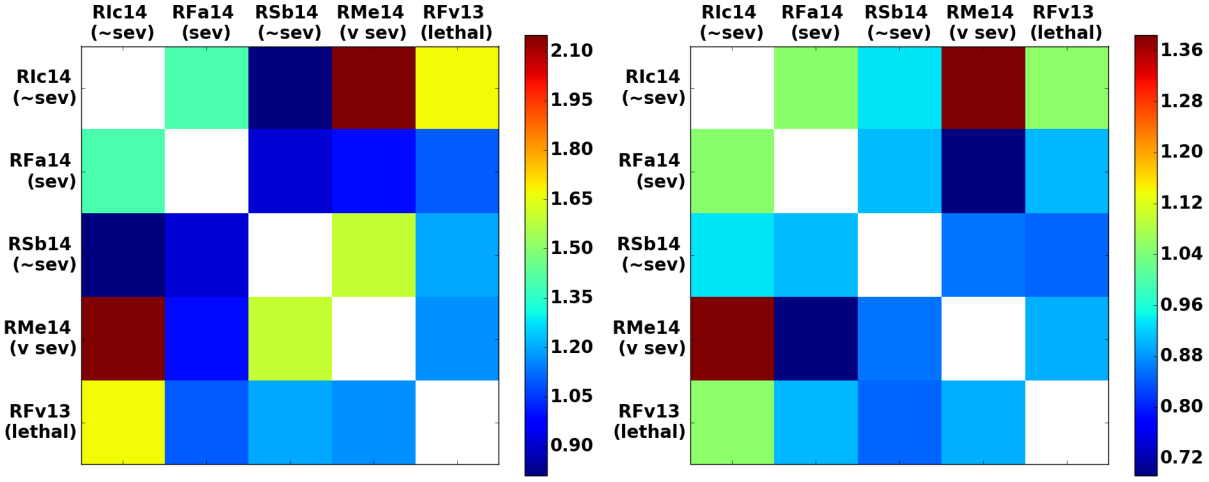


(b) Up to day 23, post-shifting on the right

FIGURE 4.15: **gran**: Comparing residual matrices, with and without Bayesian Optimization shifts.



(a) Over all days, post-shifting on the right

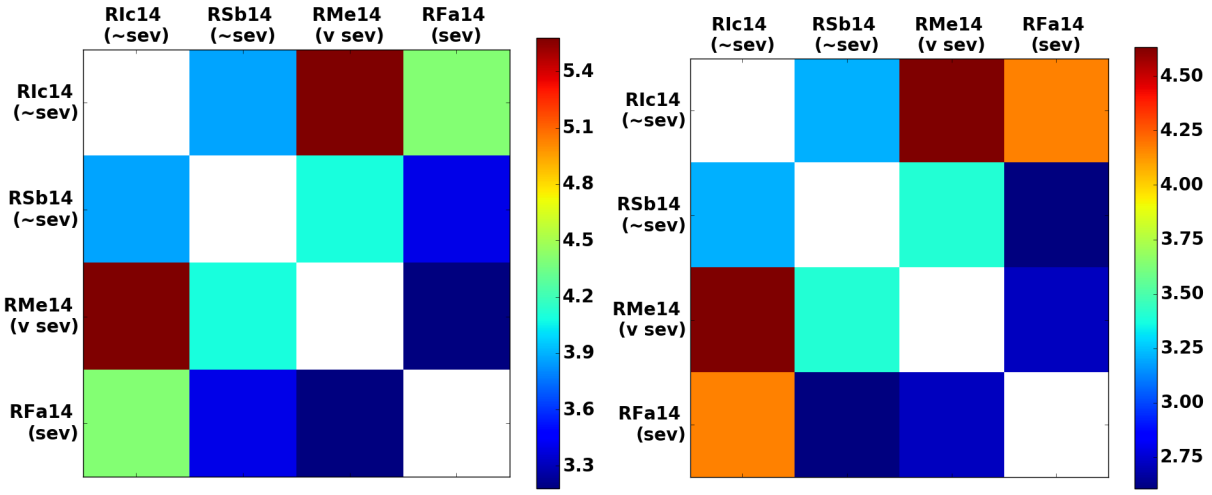


(b) Up to day 23, post-shifting on the right

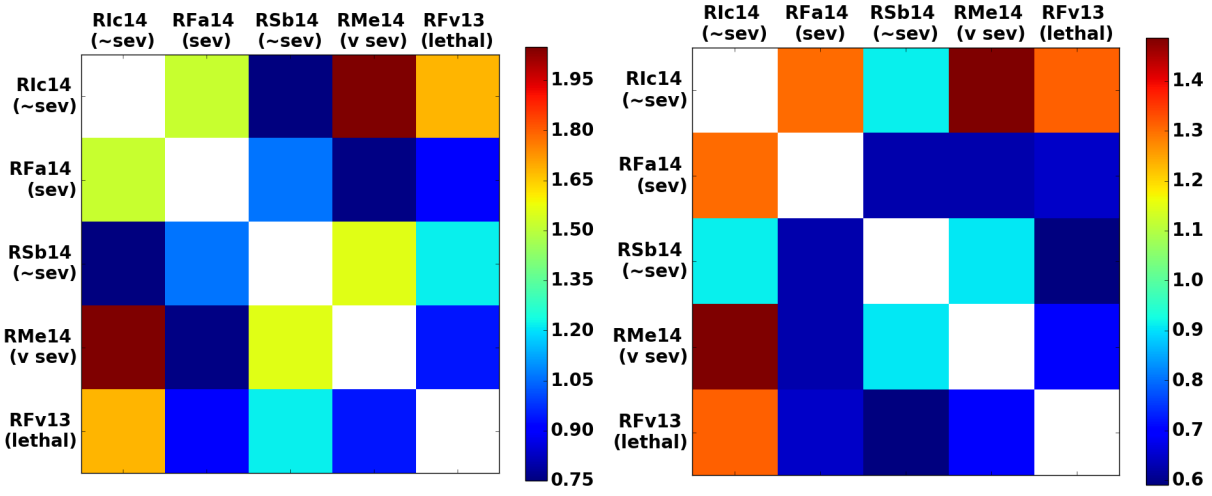
FIGURE 4.16: **hct**: Comparing residual matrices, with and without Bayesian Optimization shifts.

For the hemoglobin levels in the blood, as seen in Figure 4.17, RIc14 is least similar to the very severe monkey and most similar to the other non-severe monkey; while there are not enough monkeys to confirm the trend in general, as it does not follow as starkly for RSb14, it may show that hemoglobin is important in determining disease severity. This trend is also replicated in the first phase, since the similarity increases in the same way for RIc14 (most similar to the other

non-severe monkey and least similar to the very severe monkey). In both (a) and (b), the severe monkeys are least similar to the non-severe monkeys.



(a) Over all days, post-shifting on the right

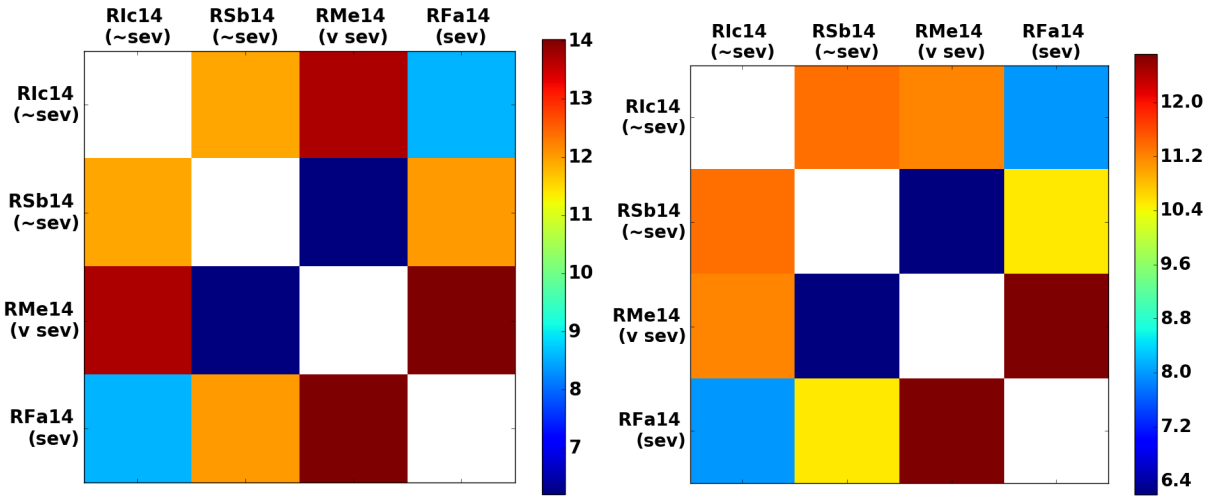


(b) Up to day 23, post-shifting on the right

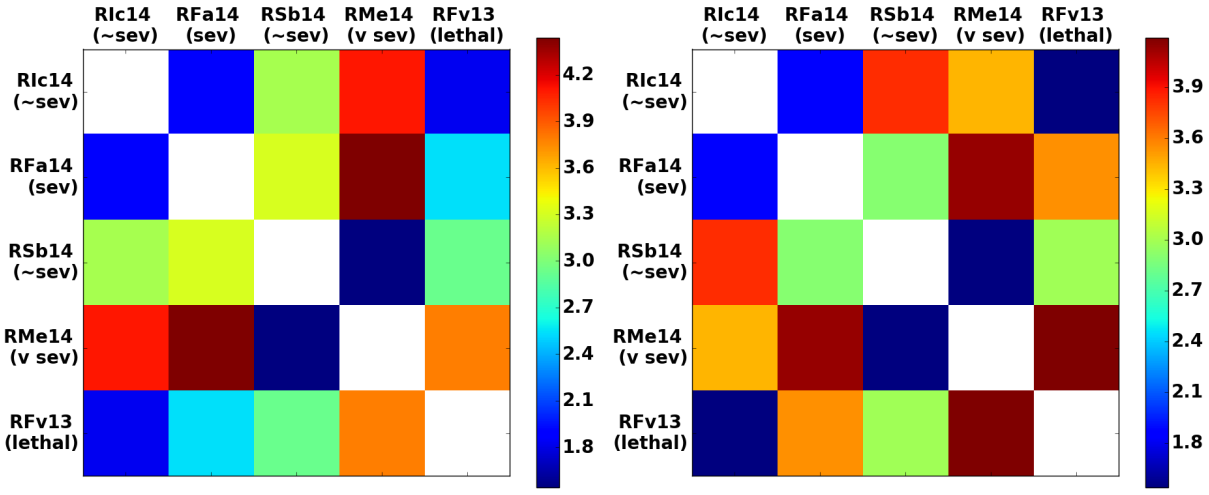
FIGURE 4.17: **hgb**: Comparing residual matrices, with and without Bayesian Optimization shifts.

Contrastingly, for lymphocytes (Figure 4.18), shifting does not exacerbate or reveal any interesting trends; the non-severe monkeys are more similar to severe monkeys and vice versa for

both over all days of the experiment and only up to day 23. The number of lymphocytes may not be a determining factor for severity all throughout the course of the infection. Regarding mean corpuscular hemoglobin, as shown in Figure 4.19, the monkeys align with respect to more similar phenotypes, except for RIc14 (non-severe) which is most similar to RMe14 (very severe). However, for up to day 23, residuals are lowest between RFa14 (severe) and RIc14 (non-severe) and between RMe14 (very severe) and RSb14 (non-severe). Mean corpuscular hemoglobin concentration is with respect to the number of red blood cells, and so it is interesting that in Figure 4.20, the two non-severe monkeys are most similar to each other in (a) but are most similar to the very severe monkey in (b). Therefore, mch and mchc, as they are related, may not be as variable in the first phase of the disease, or at least as much of an indicator, as in the next phases.

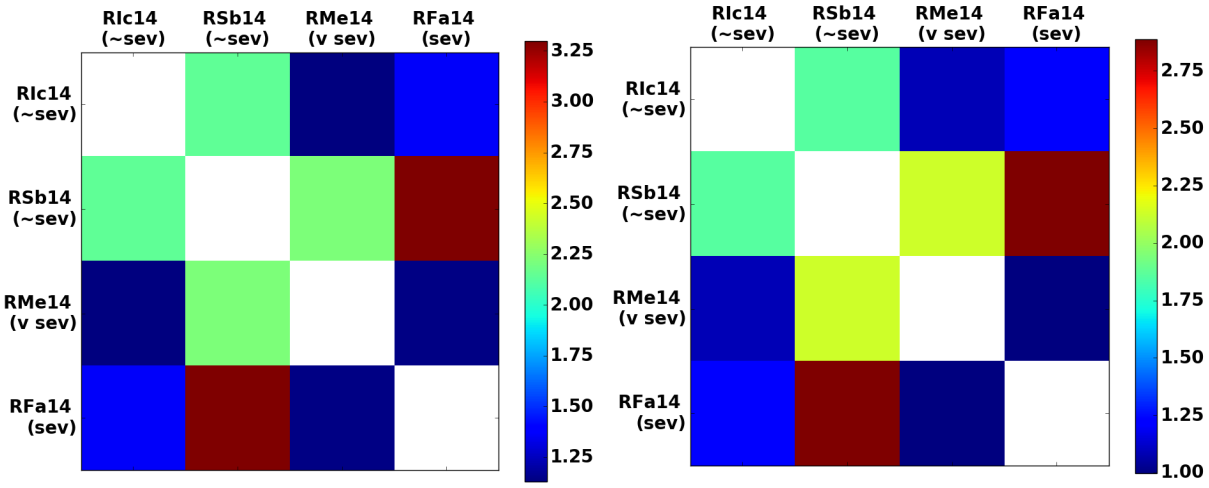


(a) Over all days, post-shifting on the right

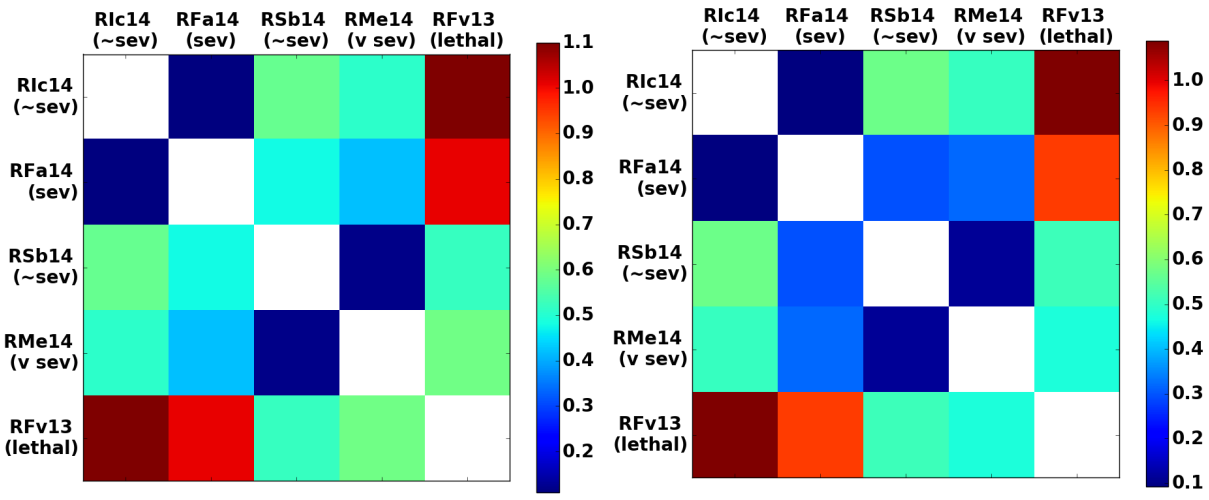


(b) Up to day 23, post-shifting on the right

FIGURE 4.18: **lymph**: Comparing residual matrices, with and without Bayesian Optimization shifts.

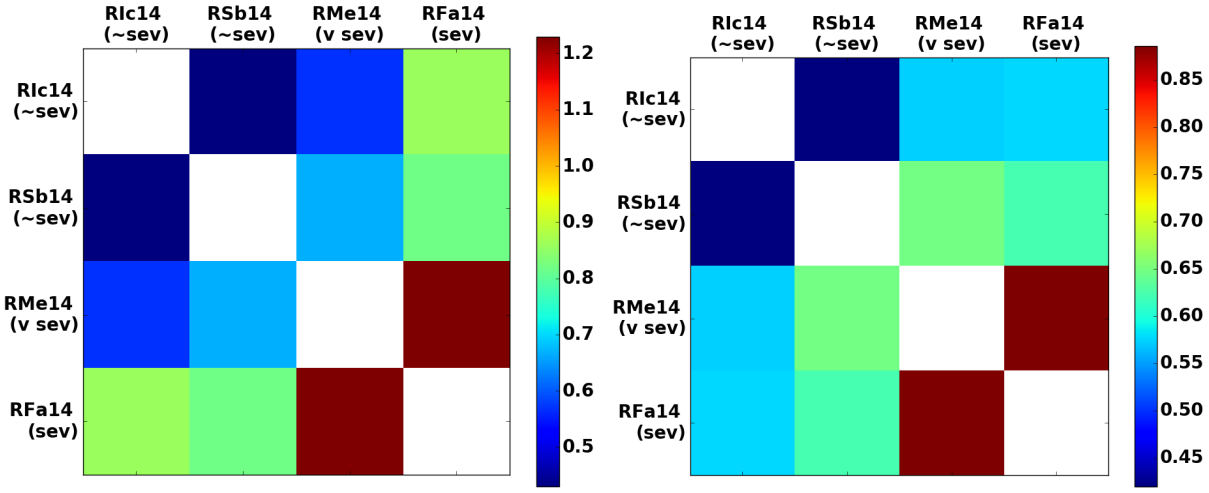


(a) Over all days, post-shifting on the right

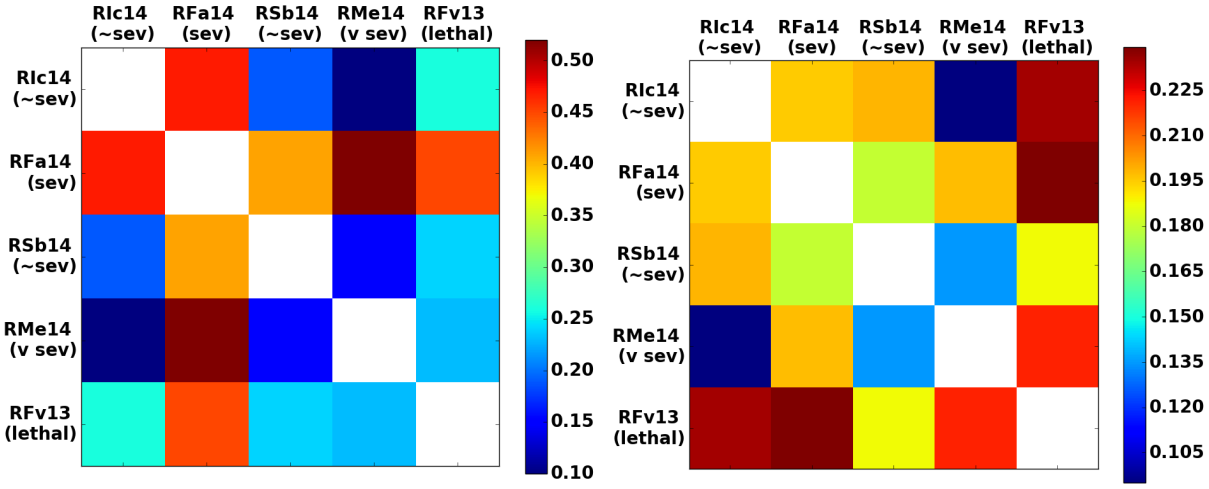


(b) Up to day 23, post-shifting on the right

FIGURE 4.19: **mch**: Comparing residual matrices, with and without Bayesian Optimization shifts.



(a) Over all days, post-shifting on the right

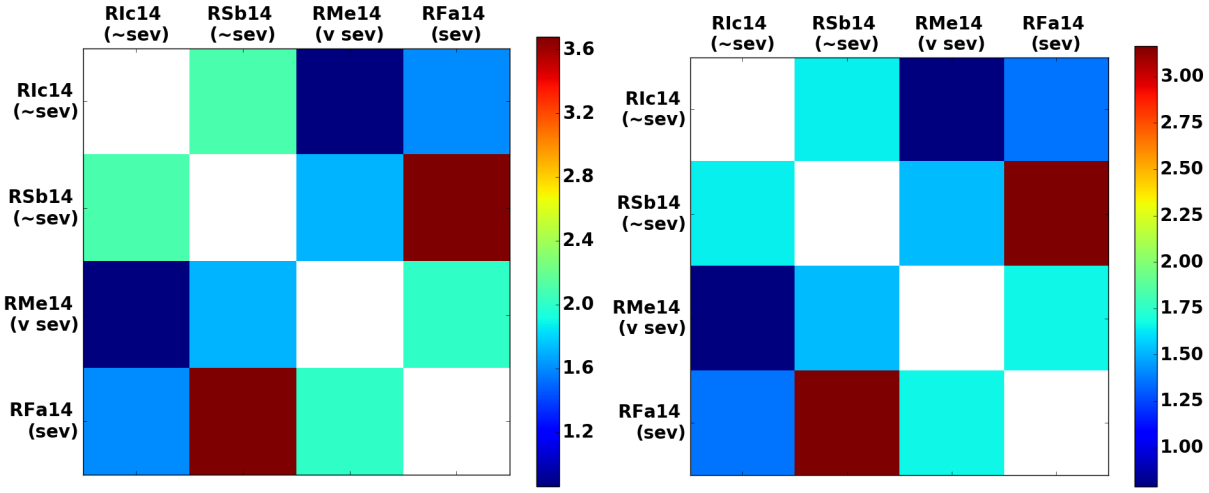


(b) Up to day 23, post-shifting on the right

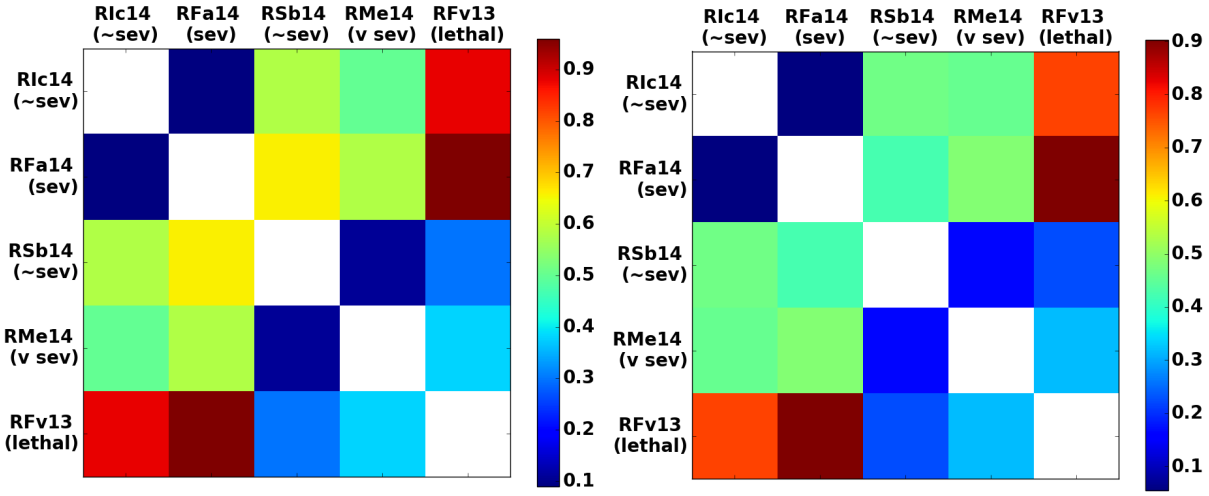
FIGURE 4.20: **mchc**: Comparing residual matrices, with and without Bayesian Optimization shifts.

The residuals of mean red blood cell volume, mcv, change very little before and after shifting, as shown in Figure 4.21, and there is again an opposing trend in both (a) and (b). These opposing trends are also seen for mean platelet volume, mpv, in Figure 4.22. The most similar residuals pairwise for mpv are between RFa14 (severe) and RIc14 (non-severe) and between

RMe14 (very severe) and RSb14 (non-severe), which are the same pairs in mean corpuscular hemoglobin for up to day 23. While the similarity between mpv and mch is telling about the relationship between hemoglobin and platelet size in the disease, both mcv and mpv may not be as indicative of severity overall based on this experiment because of the opposing trends existing in residual matrices calculated only up to day 23 and over all days (as shown in non-normalized results).

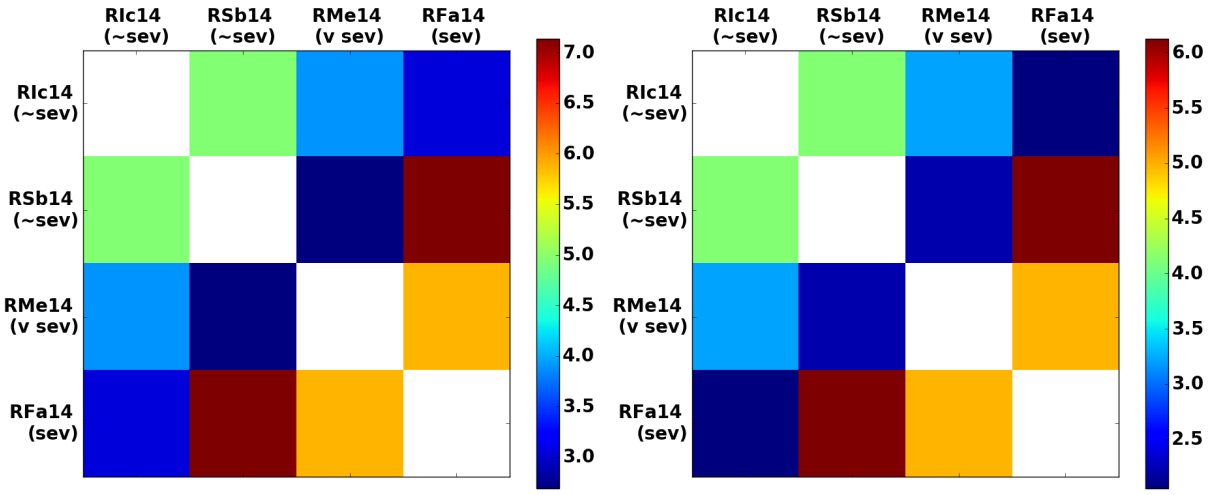


(a) Over all days, post-shifting on the right

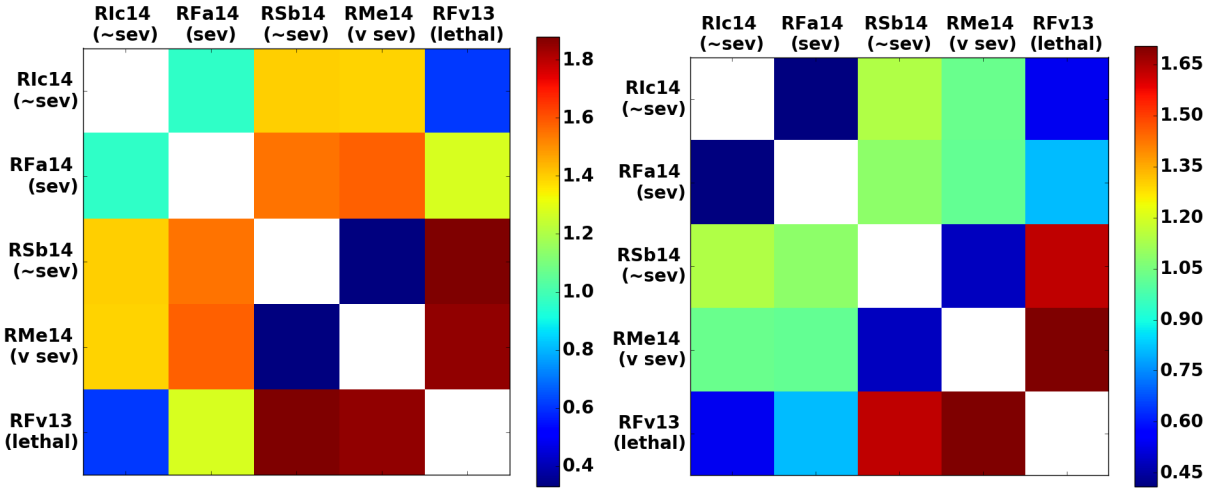


(b) Up to day 23, post-shifting on the right

FIGURE 4.21: **mcv**: Comparing residual matrices, with and without Bayesian Optimization shifts.



(a) Over all days, post-shifting on the right

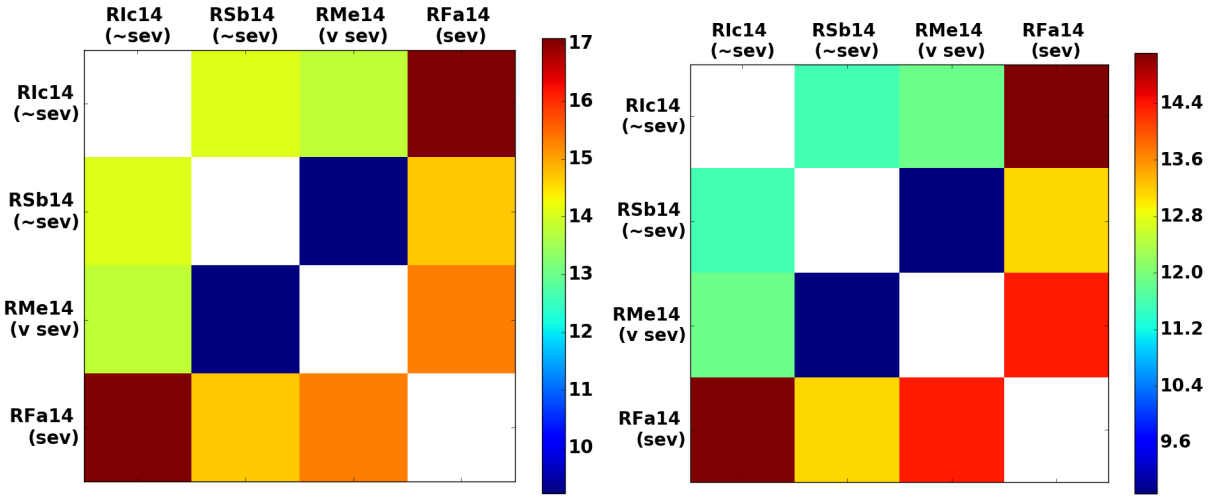


(b) Up to day 23, post-shifting on the right

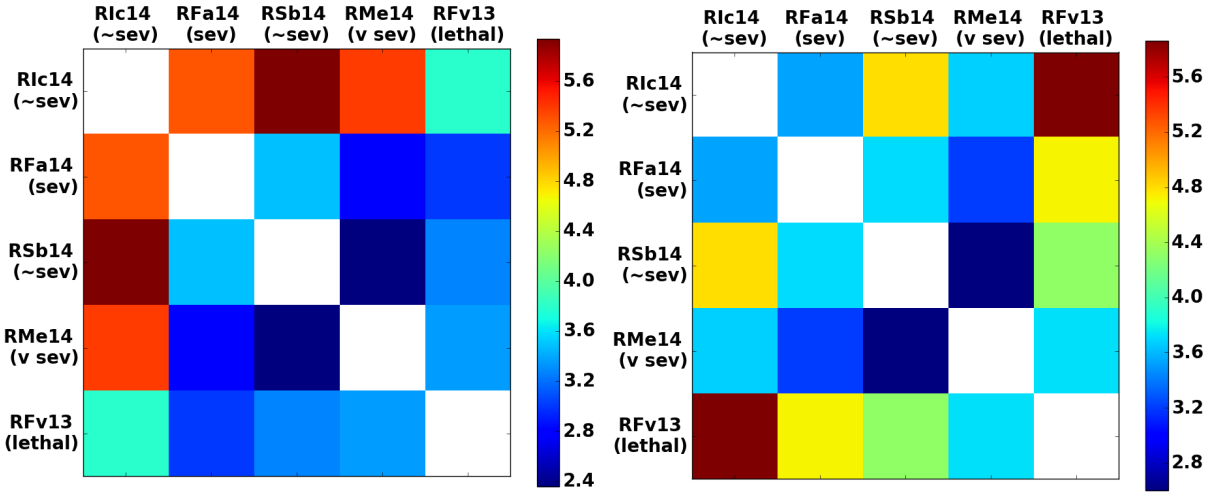
FIGURE 4.22: **mpv**: Comparing residual matrices, with and without Bayesian Optimization shifts.

Interestingly in Figure 4.23, the lethal monkey is least similar to a non-severe monkey and most similar to the very severe monkey; while the trends are not indicative over all days, this could suggest that the number of monocytes is an important trend for the first phase of the infection. In terms of the biology, monocytes are part of the innate immune system but can influence the adaptive immune system. Therefore, a better adaptive immune response could be

prognostic of survival in the long run; the monkey could respond better in relapse events, being better equipped with immunological memory at that point to curb the infection.



(a) Over all days, post-shifting on the right

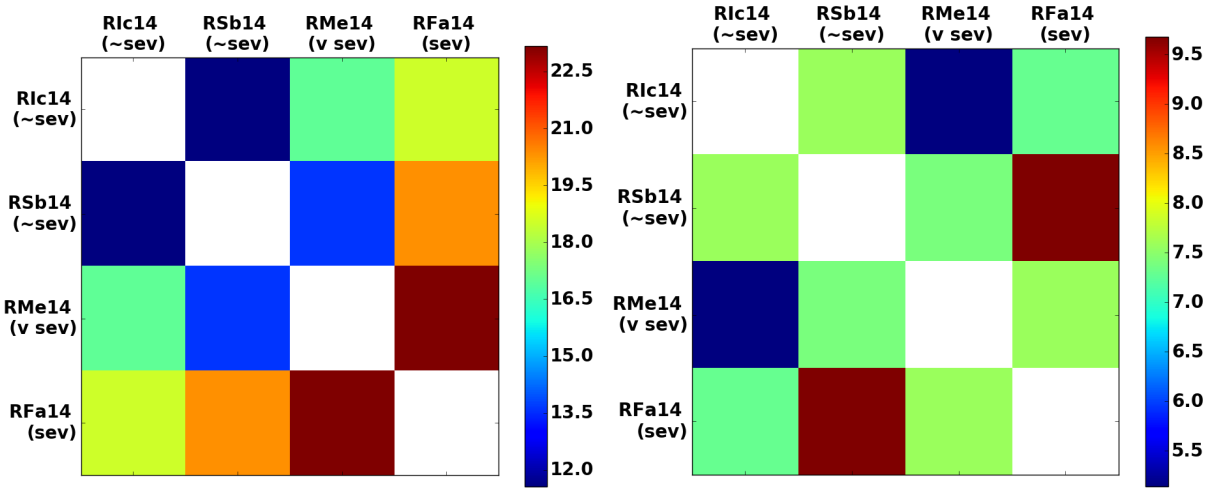


(b) Up to day 23, post-shifting on the right

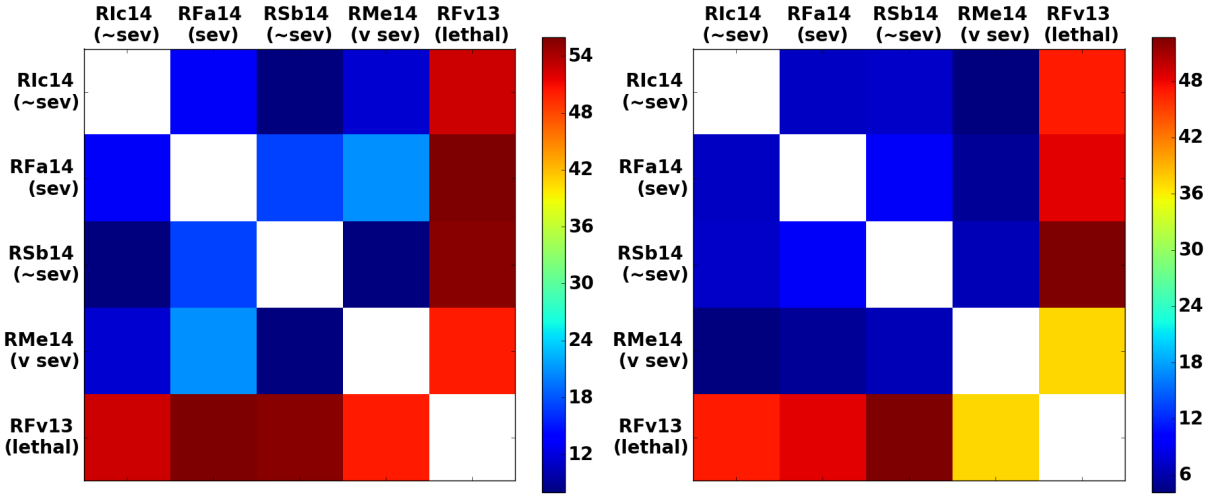
FIGURE 4.23: **mono**: Comparing residual matrices, with and without Bayesian Optimization shifts.

It does seem strange that the trend for parasitemia percent, the percentage of infected red

blood cells, is not the same as that for parasites per microliter, which reflects the overall concentration of parasites, in Figure 4.24 and Figure 4.25. Over all days, the residuals between monkeys are similar for parasitemia percent, whereas there are pronounced differences for parasites per microliter. The residual for any monkey against the lethal monkey is very different for parasitemia percent, but it is most similar to the very severe monkey. Regarding parasites per microliter, the non-severe monkeys are the most similar after shifting. Therefore, Bayesian optimization did help clarify the trends in these two clinical parameters, especially between the two non-severe monkeys over all days. However, it is also interesting that the lethal monkey, RFv13, has parasite counts most similar to the non-severe monkeys, which could indicate that it was another clinical parameter that caused the different outcome in that monkey.

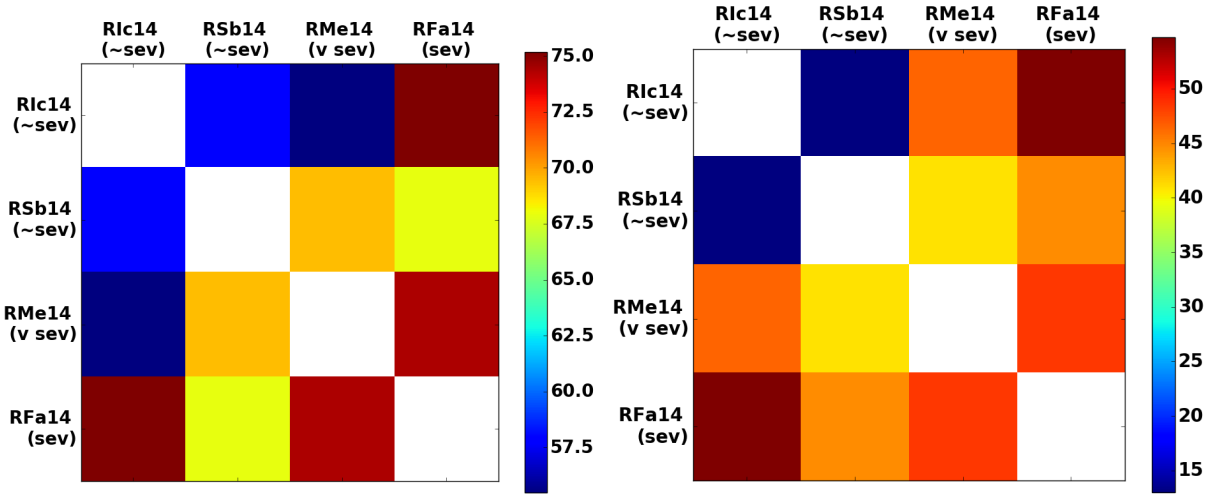


(a) Over all days, post-shifting on the right

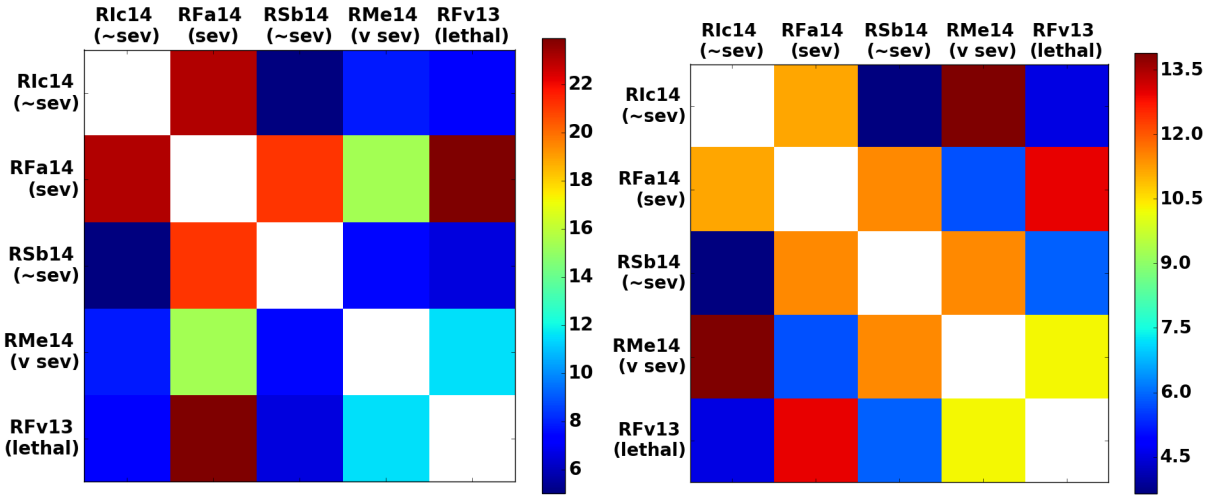


(b) Up to day 23, post-shifting on the right

FIGURE 4.24: **% parasitemia**: Comparing residual matrices, with and without Bayesian Optimization shifts.



(a) Over all days, post-shifting on the right

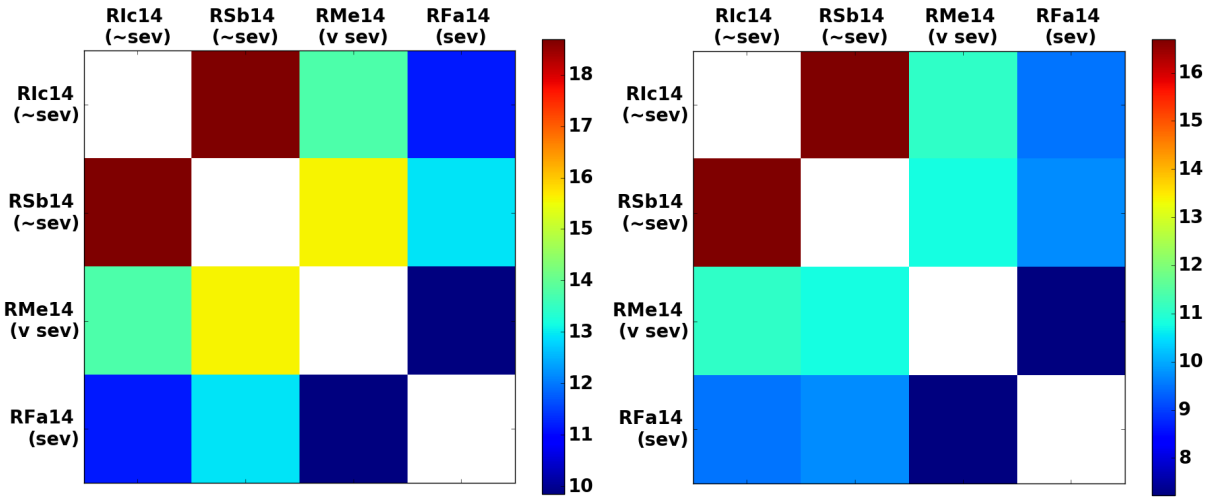


(b) Up to day 23, post-shifting on the right

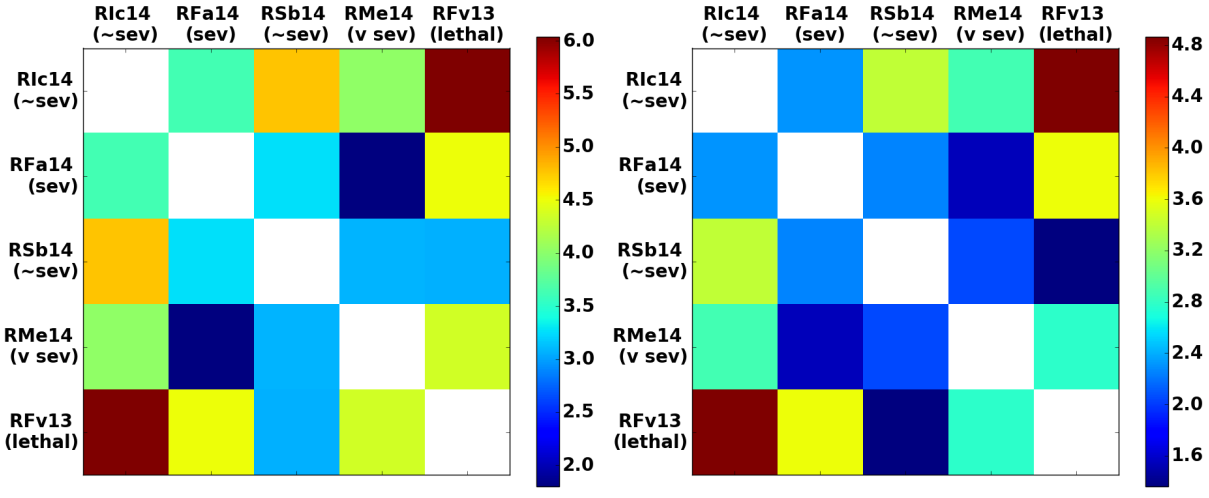
FIGURE 4.25: **parasites / uL**: Comparing residual matrices, with and without Bayesian Optimization shifts.

Regarding platelet count, in Figure 4.26, it is not exactly the same trend as mpv, even though it may seem that mean platelet volume should follow that of platelet count; instead, over all days, the severe monkeys are most similar to each other, while the two non-severe monkeys are least similar. This could mean that there is a varying platelet count in the non-severe

monkeys, while this variation is no longer present in severe phenotypes (i.e. there could be a canonical platelet count to represent more severe phenotypes, whereas there is not one for normal phenotypes). On the other hand, the results could also signify that platelet count does not determine phenotype severity.



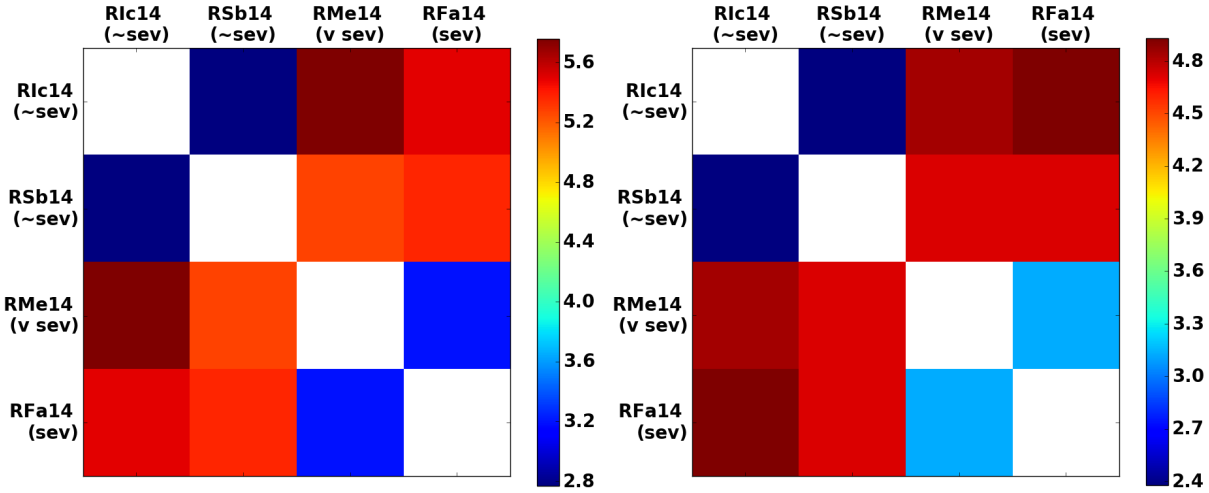
(a) Over all days, post-shifting on the right



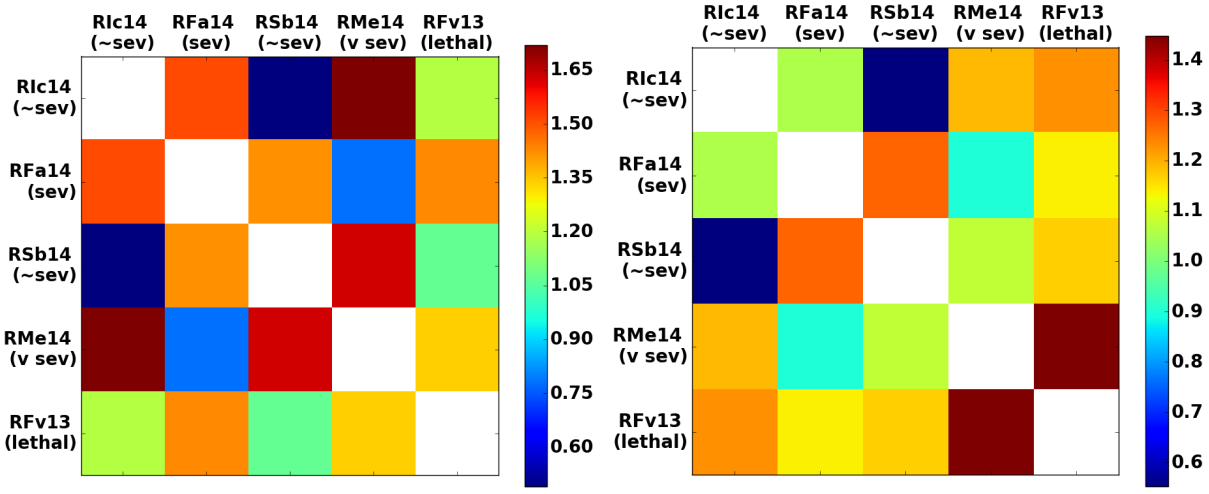
(b) Up to day 23, post-shifting on the right

FIGURE 4.26: **plt**: Comparing residual matrices, with and without Bayesian Optimization shifts.

The trends exhibited in number of red blood cells per microliter (Figure 4.27) did not change after shifting, but in general do show that the lowest residuals, in both over all days and only up to day 23, were the lowest between the two non-severe monkeys and the two severe monkeys. This makes sense, as anemia (not enough normal red blood cells) is a condition that arises with malaria.



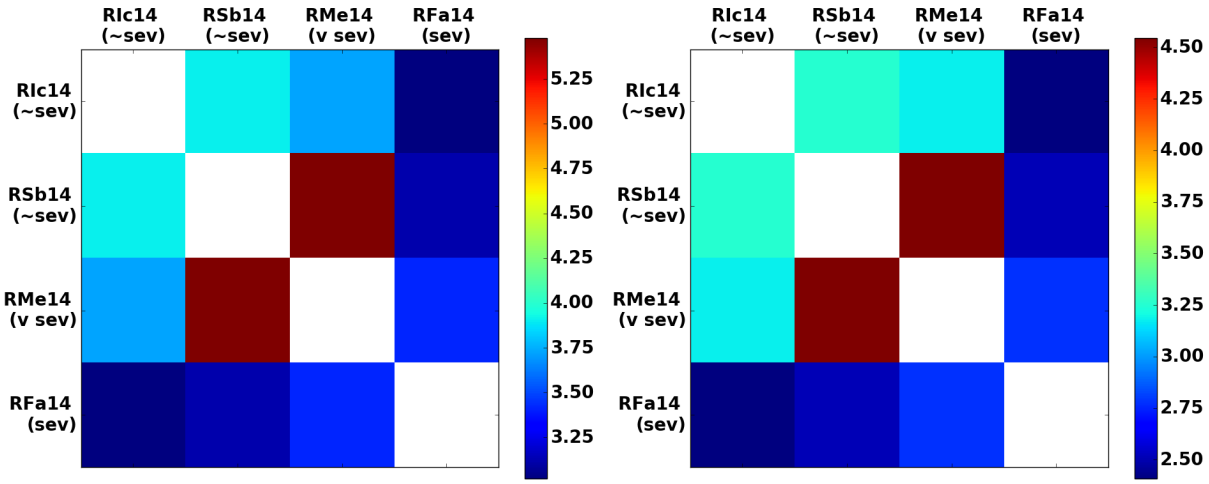
(a) Over all days, post-shifting on the right



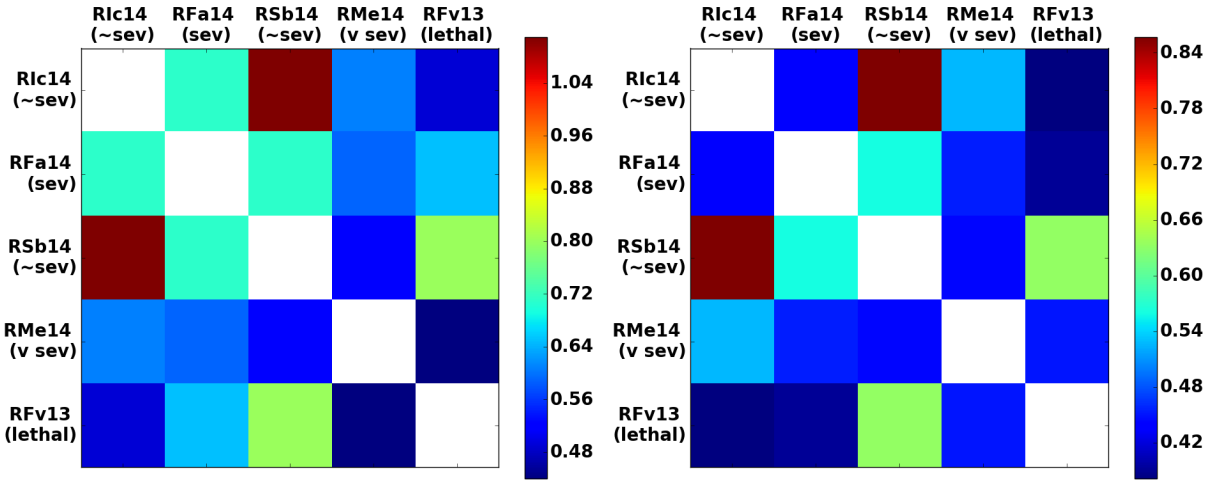
(b) Up to day 23, post-shifting on the right

FIGURE 4.27: **rbc**: Comparing residual matrices, with and without Bayesian Optimization shifts.

The red blood cell distribution width, rdw, denotes the variation in the size of red blood cells - more variation signifies illness, as a deviation from a standard size. This concept is not exactly shown in the residual matrices in Figure 4.28, although the largest residual in the matrix is between the very severe monkey and a non-severe monkey (whereas the others are a relatively similar in comparison). Therefore, while increased rdw does show illness, there could be a standard change across all phenotypes. There is not enough data to determine whether the former conclusion is acceptable, or if the the fact that the largest difference exists between the very severe and the non-severe monkey is more significant.



(a) Over all days, post-shifting on the right

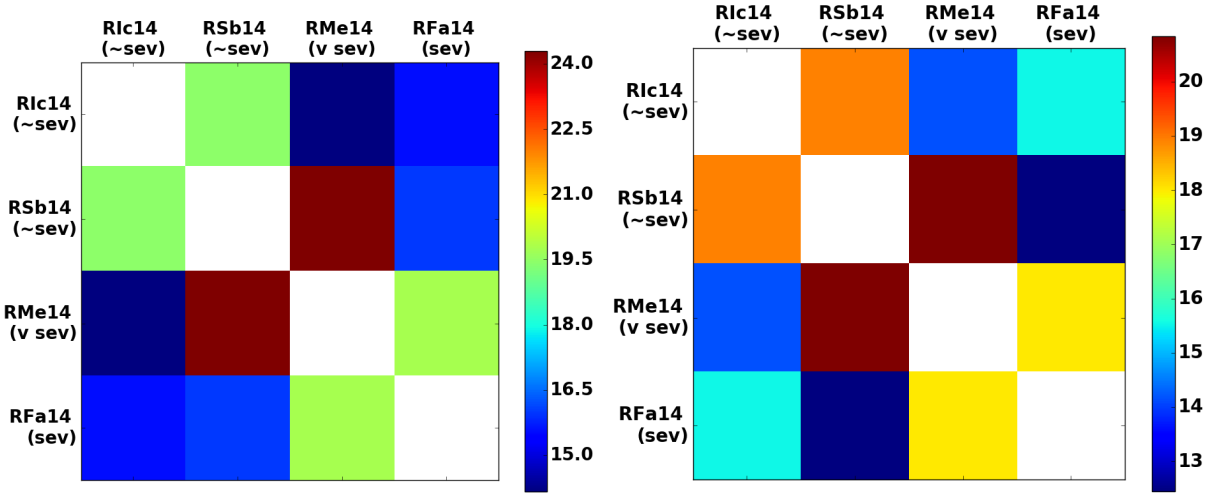


(b) Up to day 23, post-shifting on the right

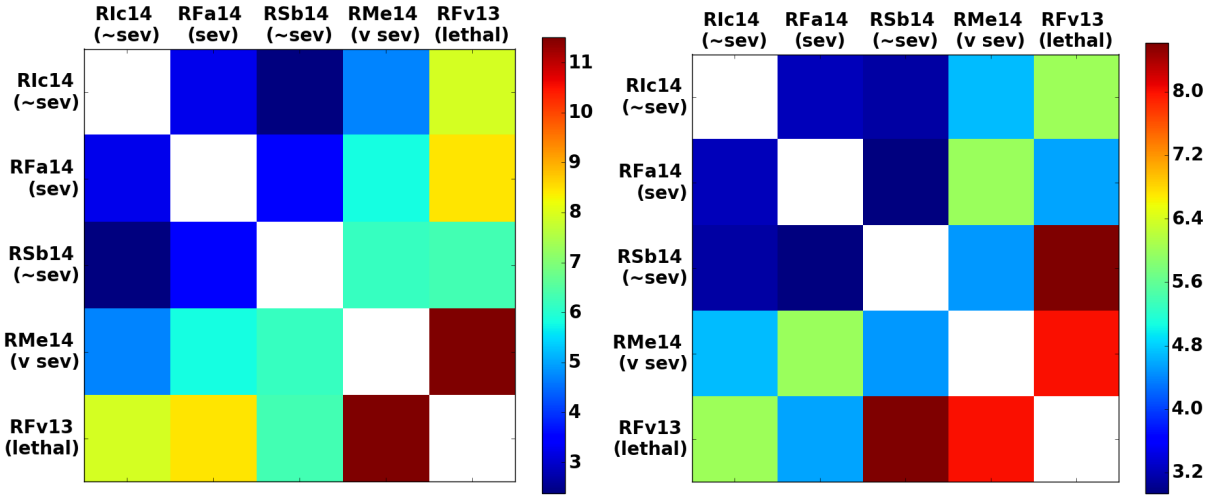
FIGURE 4.28: **rdw**: Comparing residual matrices, with and without Bayesian Optimization shifts.

Joyner et. al (Joyner et al., 2016) discuss the possible importance of reticulocyte number in survival, and this is partially shown in Figure 4.29. For up to day 23, the lethal monkey is least similar to a non-severe monkey, and likewise over all days, the very severe monkey is least similar to a non-severe monkey. Both the very severe and the lethal monkey show the greatest differences compared to other monkeys for up to day 23, thus demonstrating the importance

of the number of reticulocytes in the first phase. In addition, for white blood cell count in Figure 4.32, the lethal monkey is most similar to the two severe monkeys and least similar to the non-severe monkeys, showing that initial overall immune response, i.e. in the first phase, could be important in survival. Over all days, white blood cell count is most different between the very severe monkey and one of the non-severe monkeys, showing the same trend.

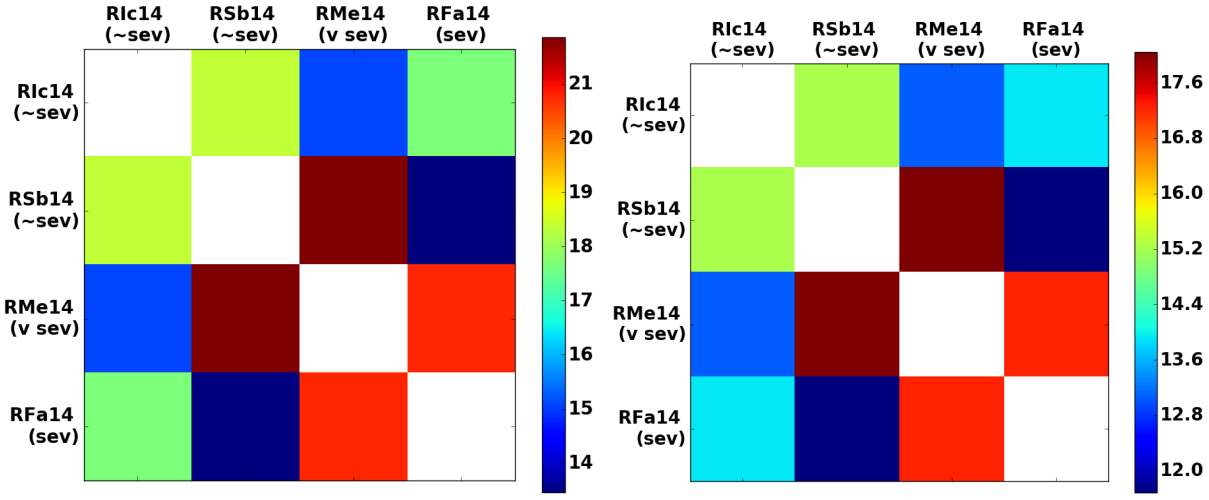


(a) Over all days, post-shifting on the right

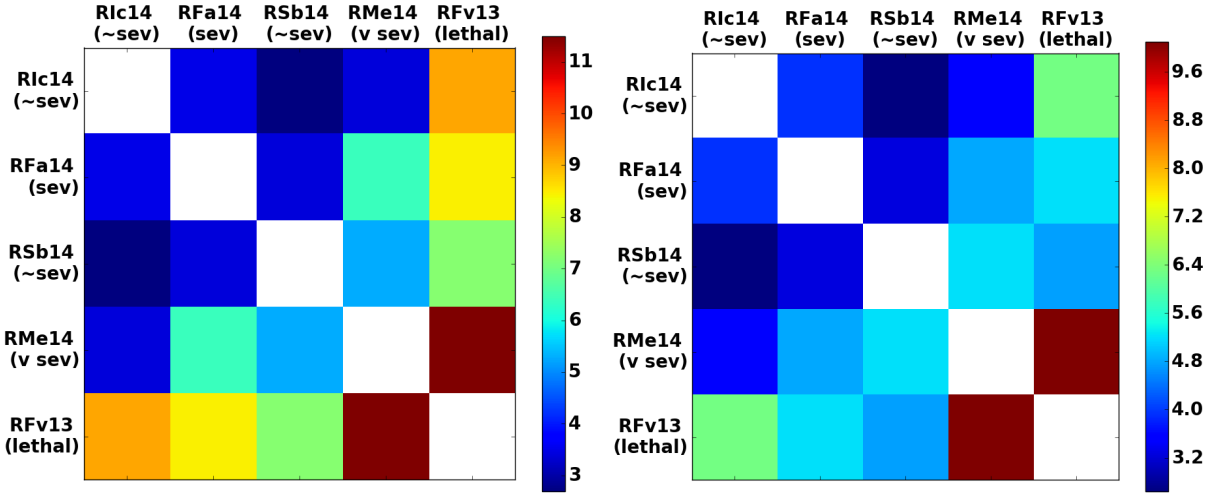


(b) Up to day 23, post-shifting on the right

FIGURE 4.29: # **reticulocytes**: Comparing residual matrices, with and without Bayesian Optimization shifts.

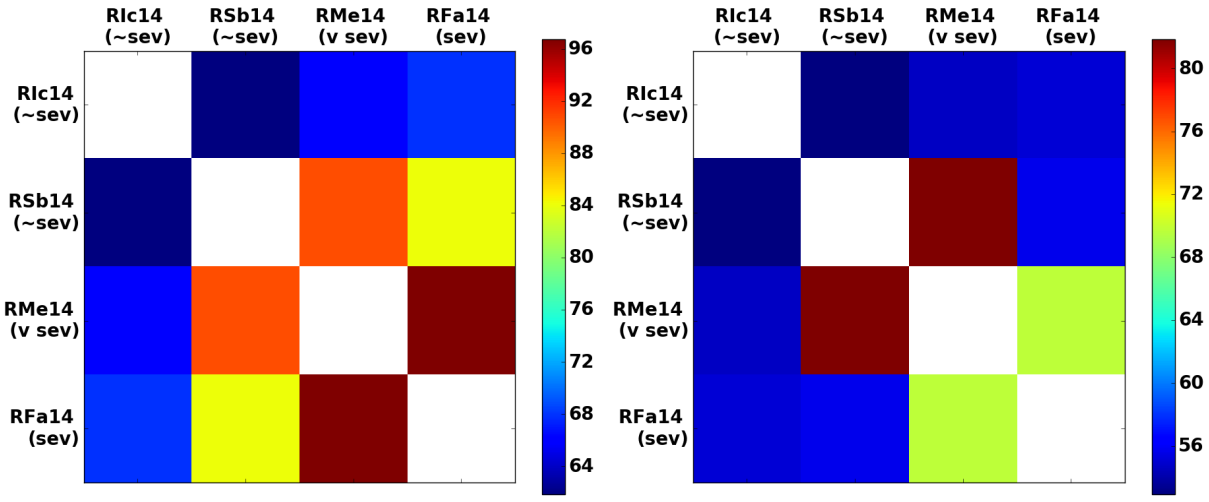


(a) Over all days, post-shifting on the right

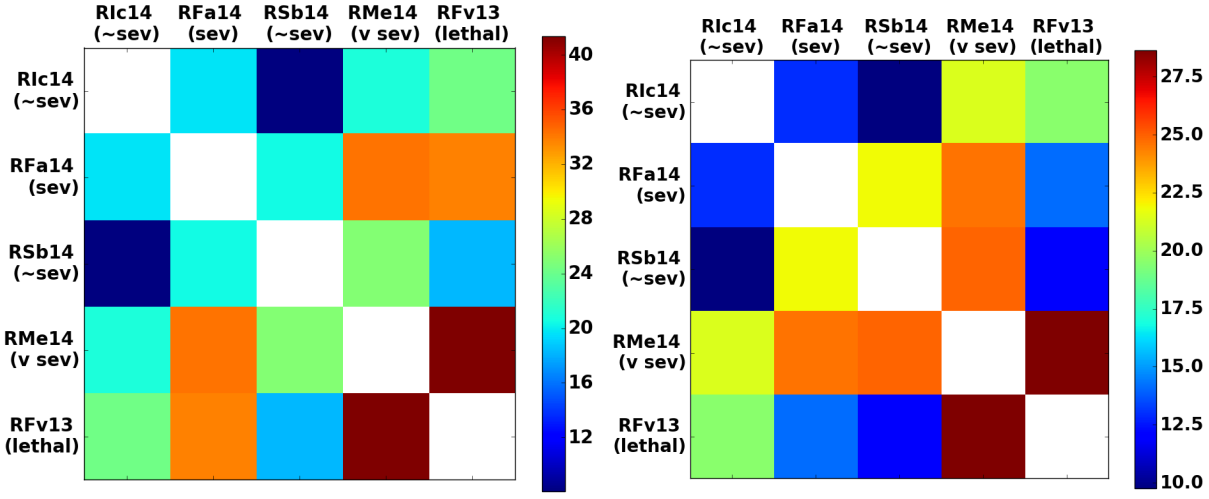


(b) Up to day 23, post-shifting on the right

FIGURE 4.30: **reticulocytes / uL**: Comparing residual matrices, with and without Bayesian Optimization shifts.

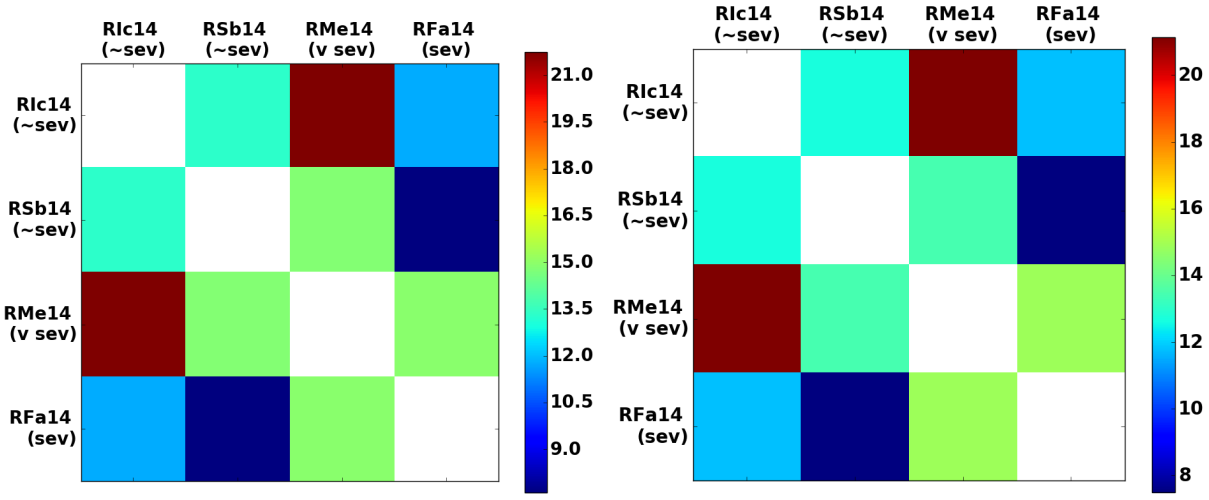


(a) Over all days, post-shifting on the right

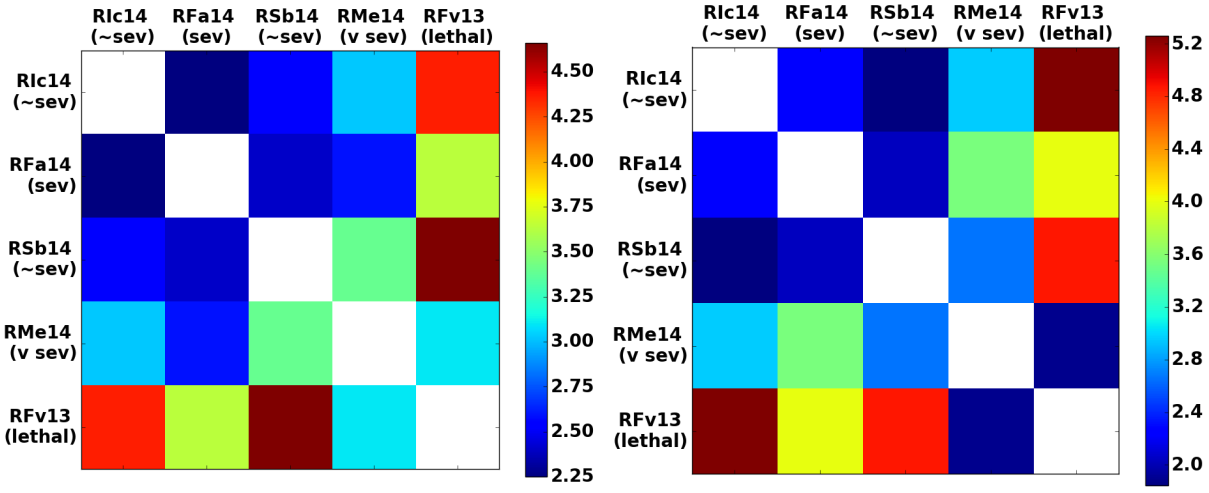


(b) Up to day 23, post-shifting on the right

FIGURE 4.31: **% reticulocytes**: Comparing residual matrices, with and without Bayesian Optimization shifts.



(a) Over all days, post-shifting on the right



(b) Up to day 23, post-shifting on the right

FIGURE 4.32: **wbc**: Comparing residual matrices, with and without Bayesian Optimization shifts.

TABLE 4.8: Quick summary of trends in residual matrices.

Clinical Parameter	Over all days (+)	Up to Day 23 (+)	Negative trend (both days)	Other*
gran	X			
hct	X			
mch	X			
mchc	X			
hgb	X	X		
rbc	X	X		
wbc	X	X		
# ret	X	X		
mono		X		
mpv			X	
mvc			X	
plt				X

* Other = severe are most similar, non-severe are least similar

(+) = severe are most similar to each other, non-severe are most similar to each other

(-) = severe are most similar to non-severe, and vice-versa

TABLE 4.9: Summary of analyses

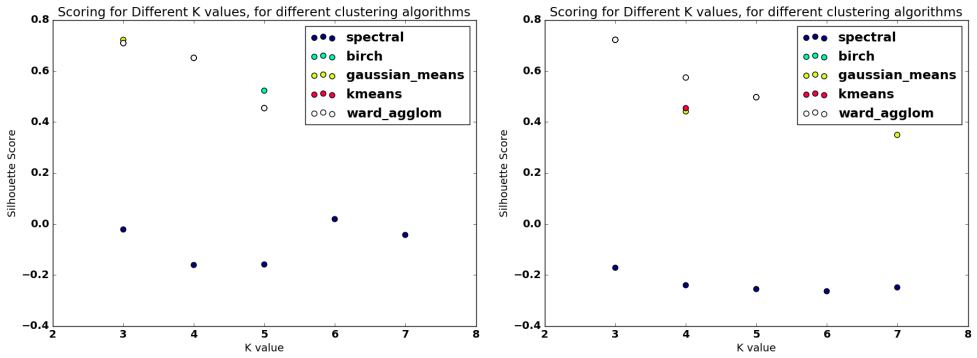
Joyner et al. (2016)	Automated analyses
# Reticulocytes - possible indicator of lethal phenotype	X
Anemic phenotype worsens with severity	X
Relationship between hemoglobin and: parasitemia kinetics, mean corpuscular volume (red blood cell size)	X
Role of thrombocytopenia (platelet deficiency) not well understood	X
Lower parasitemia in non-severe phenotype	X

4.3.1.1 Clustering Results

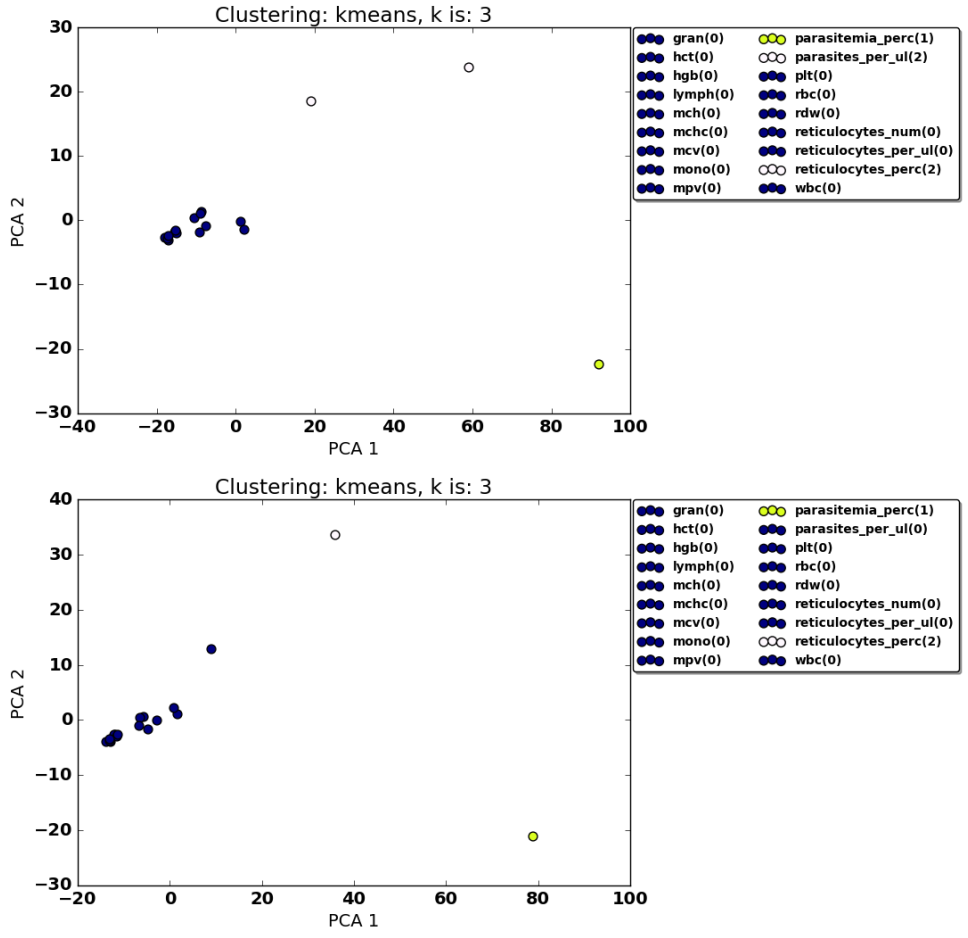
For Figure 4.33, the silhouette score stayed relatively constant after $k = 3$ (where k is the number of clusters), and thus the clusters for $k = 3$ and $k = 4$ are shown because at $k = 2$ the result would not be meaningful (since two parameters are very far away from the others). At $k = 3$, it is interesting that after shifting, parasites per microliter are clustered with reticulocytes and monocytes, while this data-point is in its own cluster pre-shifting; this change suggests that for up to day 23, reticulocytes and monocytes could be important in determining the severity of the disease, as previously discussed in Section 4.3.1. At $k = 4$, in contrast, the clusters are the same both pre- and post-shifting. The clusters do make sense in terms of the biology; the white blood cell types are in the same cluster, while the more red-blood cell related parameters are in

another. Platelets are separate from mean platelet volume, which makes sense in that platelet size does not necessarily correspond with platelet number.

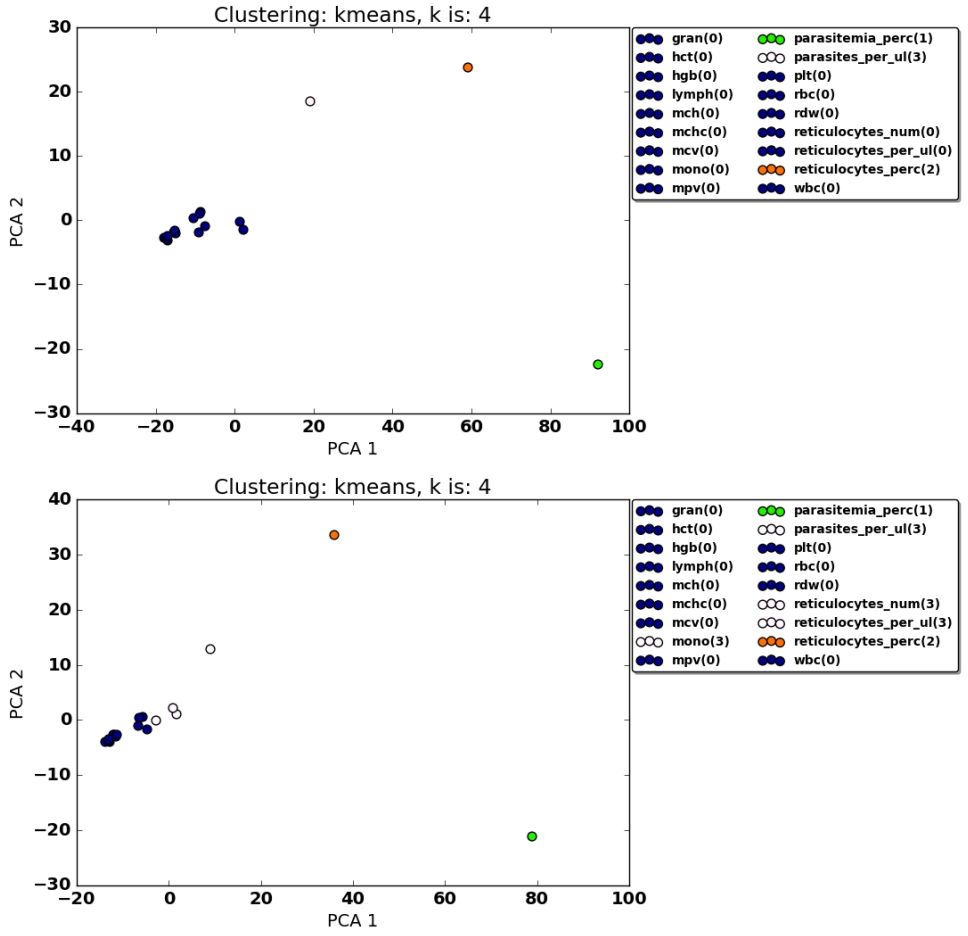
The clusters over all days, as shown in Figure 4.34, are similar to those found in up to 23 days and are the same for both pre- and post-shifting. However, as seen at $k = 3$, parasites per microliter are grouped with reticulocytes percentage, rather than number or concentration; the percentage reflects the proportion of reticulocytes out of total red blood cells (rather than raw count or concentration). It is unclear if this result has biological significance or if because the two outlying 'clusters' are slightly closer together, they are grouped together. Either way, the previous finding that reticulocyte count and concentration clustered with parasites per microliter is consistent with the conclusion in the paper- that a critical window of a slight increase in reticulocytes in the first phase could have been an indicator of the lethal monkey phenotype (Joyner et al., 2016); the fact that it does not show in the clustering over all days also matches this idea, since the critical window is early.



(a) Silhouette scores, post-shifting on the right

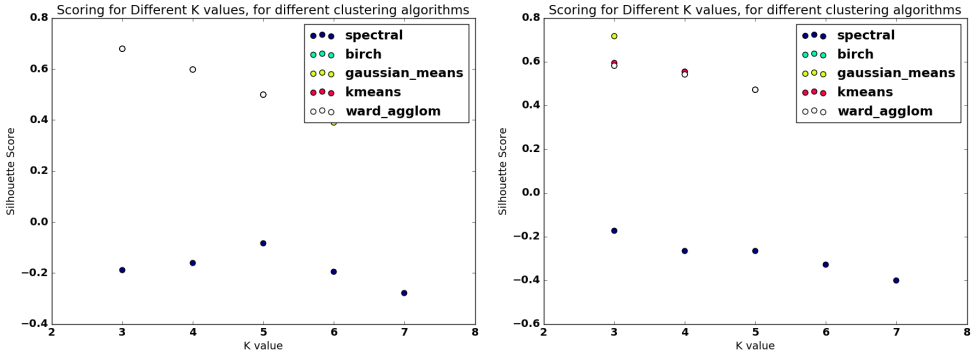


(b) $k = 3$, post-shifting on the bottom

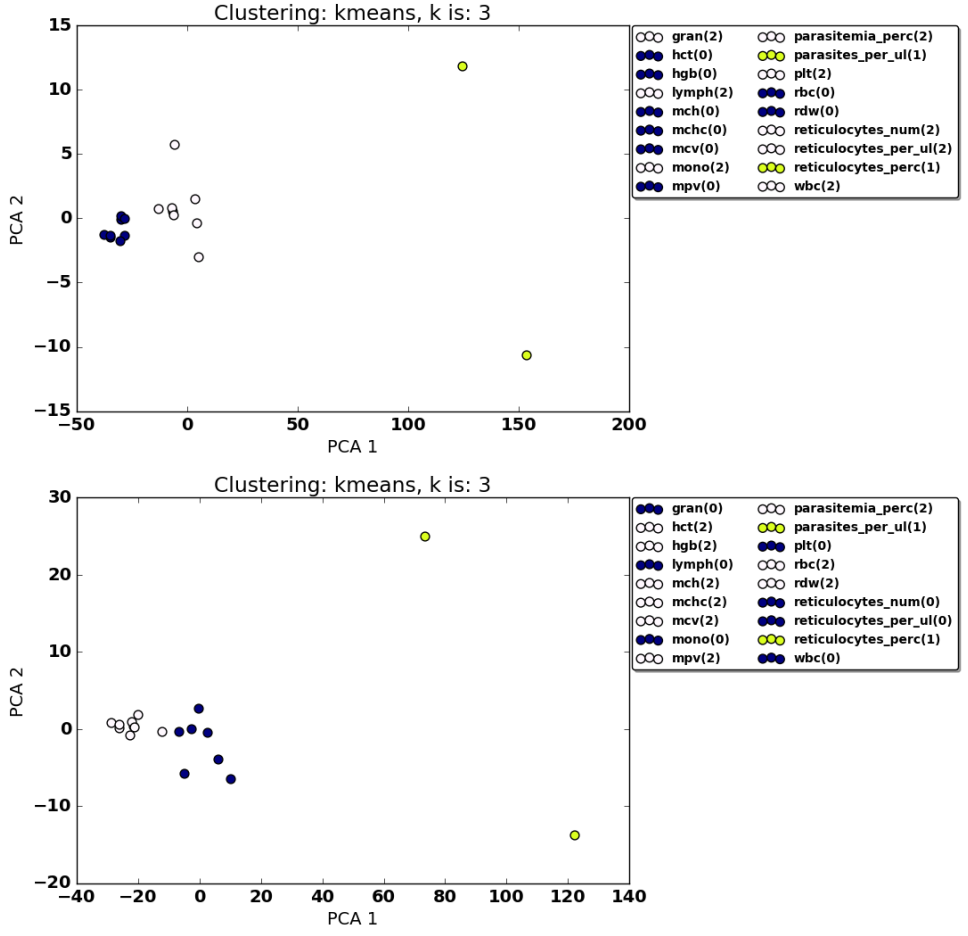


(c) $k = 4$, post-shifting on the bottom

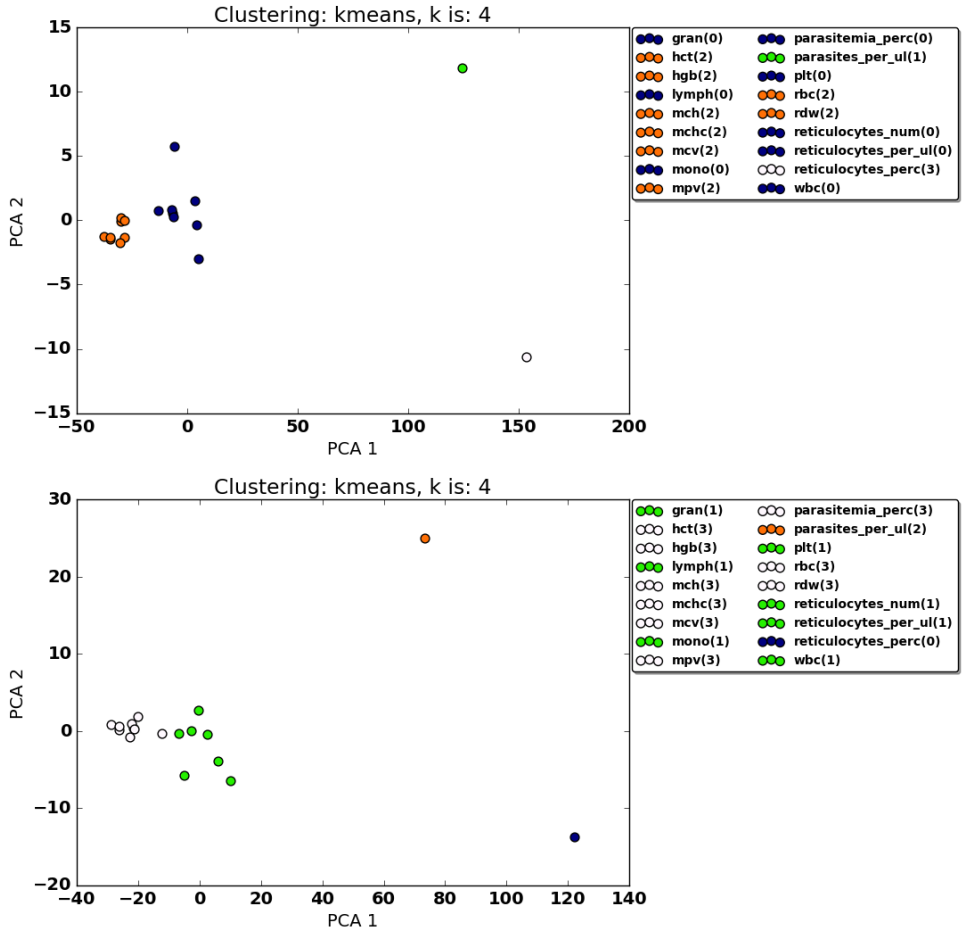
FIGURE 4.33: Up until day 23 (including RFv13), clustering the residual matrices to characterize each clinical parameter.



(a) Silhouette scores, post-shifting on the right



(b) $k = 3$, post-shifting on the bottom

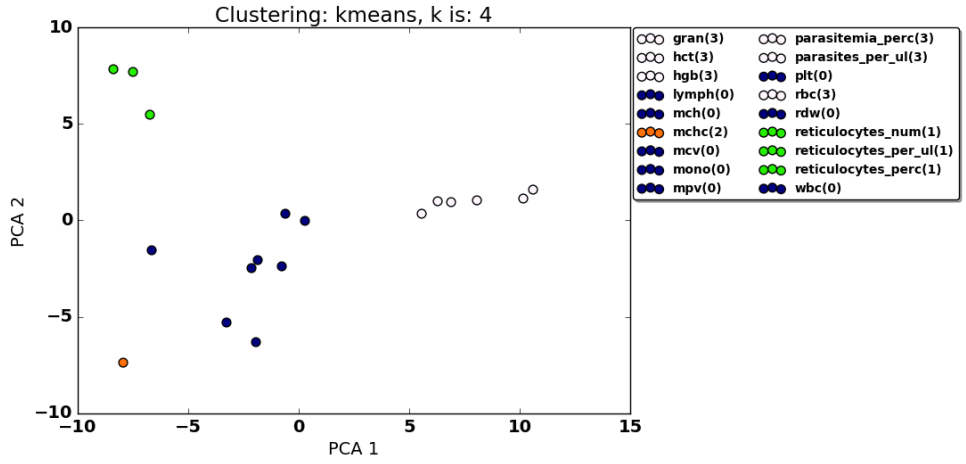


(c) $k = 4$, post-shifting on the bottom

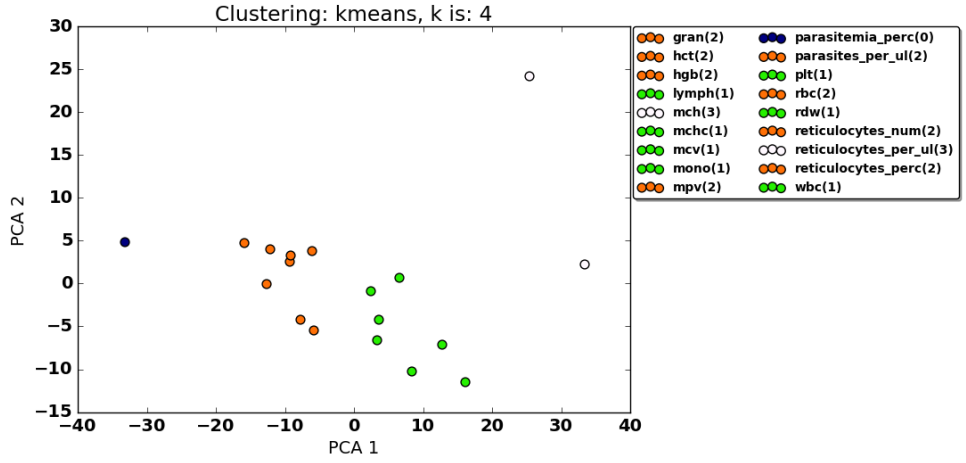
FIGURE 4.34: Over all days (excluding RFv13), clustering the residual matrices to characterize each clinical parameter.

Looking at the normalized clusters was also interesting, as seen in Figure 4.35. Clustering non-normalized data resulted in clinical parameters that were grouped in terms of similar cell types, i.e. clustering red blood cell related parameters together and white blood cell related parameters together. The normalized results instead show clusters representative of different information. For example, platelets are clustered with mean corpuscular volume for both up to day 23 and over all days, which associates thrombocytopenia with anemia. While both conditions are signs of malaria, the role of thrombocytopenia in disease severity is debated and uncertain (Joyner et al., 2016). Thus, the normalized clusters could be gleaning information about the

relationship between the two conditions, since the residuals between monkeys are similar for both. Moreover, the parasitemia level is grouped specifically with granulocytes, hematocrit, hemoglobin, and red blood cell count over 23 days, while the same cluster includes reticulocytes and mean platelet volume over all days. This cluster composition change shows the importance of different stages in the longitudinal infection. Overall the normalized results not only support the finding in Joyner et al. (2016) that reticulocyte count is significant in determining disease severity, but also suggest other parameters that could be indicative of infection stage or severity.



(a) $k = 4$, up to day 23



(b) $k = 4$, over all days

FIGURE 4.35: Comparing normalized results for up to day 23 (excluding RFv13) and over all days.

4.3.2 Sign-Match Matrices

Sign-match matrices show similar results to that of normal clustering, in that many of the data-points cluster better post-shifting (even as silhouette scores are lower, the representation in space is more reasonable). At first the patterns in the pre/post shifting matches do not seem relevant; enough parameters change signs before and after shifting such that the total number of matches before and after are very similar. As an example, the two non-severe monkeys, RIc14 and RSb14, matched at 10 signs before shifting and only 8 after; this reduction seems counter-intuitive because the non-severe monkeys are expected to have an increase in matches, reflective of the residual matrices results. However, it is shown in clustering that Bayesian Optimization improves representation in vector space and therefore analysis, even if the sign matches seem contrary at first; these changes suggest that some sign matches could be more important than others.

TABLE 4.10: Signs that matched in regression model coefficients, comparing monkeys pairwise, all days (exclude RFv13); pre = pre-shifting, post = post-shifting.

	RIc14_RFa14		RIc14_RSb14		RIc14_RMe14		RFa14_RSb14		RFa14_RMe14		RSb14_RMe14	
	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post
rdw	1	1	1	1	1	0	1	1	1	1	1	1
plt	0	0	1	1	1	1	0	0	0	1	1	0
hct	0	1	1	0	0	1	0	1	1	1	0	0
mch	1	0	1	0	0	1	1	0	0	1	1	0
ret / uL	0	0	0	1	0	0	1	1	1	1	1	1
mchc	1	1	1	0	1	0	1	1	1	1	1	0
# ret	1	1	1	0	1	1	1	1	1	1	1	1
mpv	1	1	1	0	1	0	1	1	1	1	1	1
rbc	1	0	1	1	1	0	1	1	1	0	1	1
ret %	1	1	1	1	1	1	1	1	1	1	1	1
wbc	0	0	0	1	0	1	1	1	1	1	1	0
gran	1	1	0	0	1	1	0	1	1	0	0	1
lymph	1	0	1	0	0	0	1	1	0	0	0	1
mono	0	1	1	1	1	0	0	1	0	0	1	0
hgb	1	1	0	1	1	0	0	0	1	1	0	0
mcv	1	1	1	1	1	1	1	1	1	1	1	1
sum	11	10	12	9	11	8	11	13	12	12	12	9

4.3.2.1 Kmeans Clustering

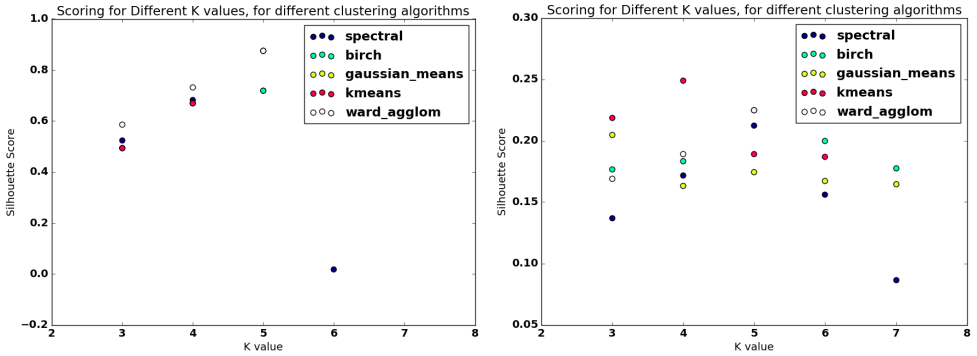
As seen in both Figure 4.36 and Figure 4.37, the silhouette scores of the matrices are much lower after shifting, so the clusters are scored better before shifting (but seem more reasonable after shifting). The cluster composition, for the shifted clustering, is not very different than that for residual matrices, as there are only a few parameters that switch groups. For example, in Figure 4.37, there is now a cluster for number of reticulocytes, hemoglobin, and monocytes, which reflects the previous finding that in the first phase, reticulocytes and monocytes could be

TABLE 4.11: Signs that matched in regression model coefficients, comparing monkeys pairwise, up to day 23 (include RFv13); pre = pre-shifting, post = post-shifting.

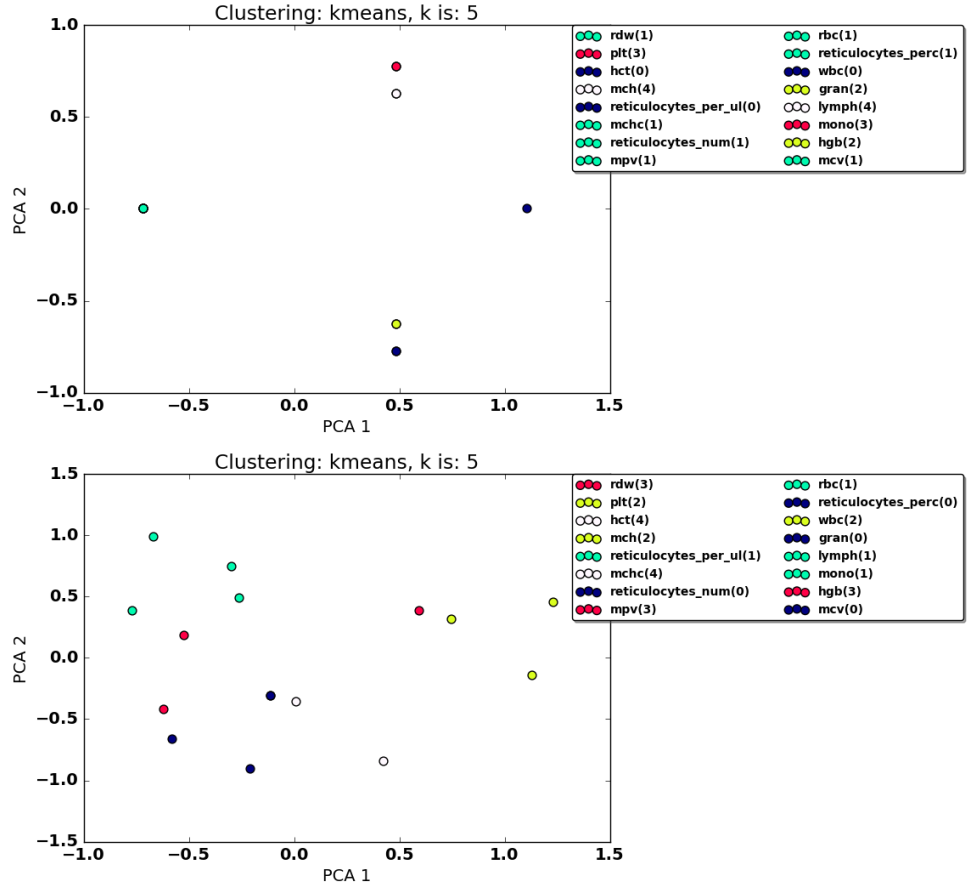
	RIc14_RFv13		RIc14_RSb14		RIc14_RMe14		RIc14_RFa14		RFv13_RSb14	
	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post
rdw	0	1	1	0	1	1	1	0	0	0
plt	0	0	0	1	1	1	1	1	1	1
hct	1	0	0	1	1	1	1	1	0	1
mch	1	1	1	0	1	0	1	0	1	1
ret / uL	0	1	1	1	0	1	1	1	0	0
mchc	1	0	1	0	1	0	1	0	1	1
# ret	1	1	1	1	1	1	1	0	1	1
mpv	1	1	1	0	1	0	1	1	1	1
rbc	0	1	0	0	1	0	0	0	1	1
ret %	1	0	1	0	1	1	1	1	1	1
wbc	0	0	0	0	1	0	1	1	1	1
gran	1	1	0	0	0	0	0	1	0	1
lymph	0	1	1	1	1	1	0	0	0	0
mono	0	0	0	1	0	1	0	0	1	1
hgb	0	1	1	1	1	0	1	1	0	1
mcv	0	1	1	1	1	1	1	1	0	0
sum	7	10	10	8	13	9	12	9	9	12

	RFv13_RMe14		RFv13_RFa14		RSb14_RMe14		RSb14_RFa14		RMe14_RFa14	
	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post
rdw	0	0	0	0	1	1	1	0	1	1
plt	0	1	0	1	0	1	0	1	1	0
hct	1	0	1	0	0	1	0	1	1	1
mch	1	1	1	0	1	0	1	1	1	0
ret / uL	0	1	0	0	0	1	1	1	0	1
mchc	1	1	1	1	1	1	1	1	1	1
# ret	1	1	1	1	1	0	1	0	1	0
mpv	1	1	1	1	1	1	1	0	1	0
rbc	0	0	1	1	0	1	1	1	0	0
ret %	1	1	1	0	1	1	1	1	1	1
wbc	0	1	0	1	0	1	0	0	1	1
gran	0	1	0	1	1	1	1	1	1	1
lymph	0	1	1	0	1	1	0	1	0	1
mono	1	0	1	0	1	0	1	0	1	0
hgb	0	1	0	1	1	0	1	0	1	0
mcv	0	0	0	0	1	1	1	1	1	0
sum	8	10	9	8	11	12	12	10	13	8

important indicators. Hemoglobin is normally found to be negatively correlated with parasite number, so it is interesting that it clusters with the other two parameters; this change could be due to the fact that while some clinical parameters may be positively correlated and others negatively correlated with parasite counts, these data-points clustered together because of similar significance in predicting infection severity. Over all days, platelets cluster with mean corpuscular hemoglobin and notably white blood cell count in Figure 4.36, possibly showing the relationship between increased white blood cell count (i.e. greater immune response) and thrombocytopenia (i.e. a sign of illness, a low count of platelets). Therefore, this type of clustering could be pulling out different information about the relationships among clinical parameters, as compared to the residual matrices, even as shifting does lower silhouette scoring.

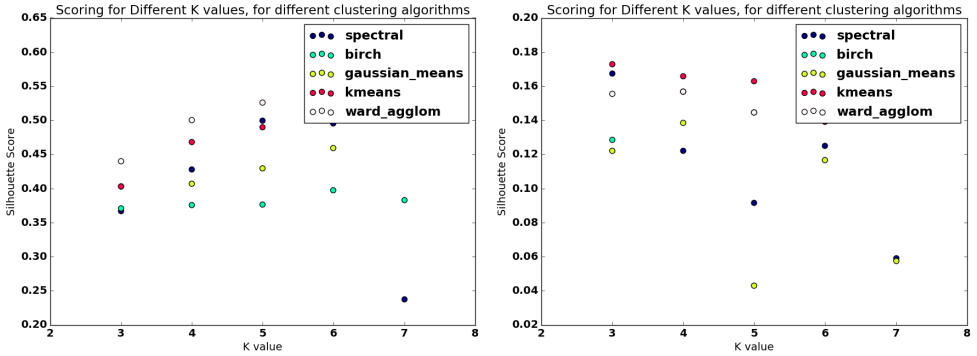


(a) Silhouette scores, post-shifting on the right

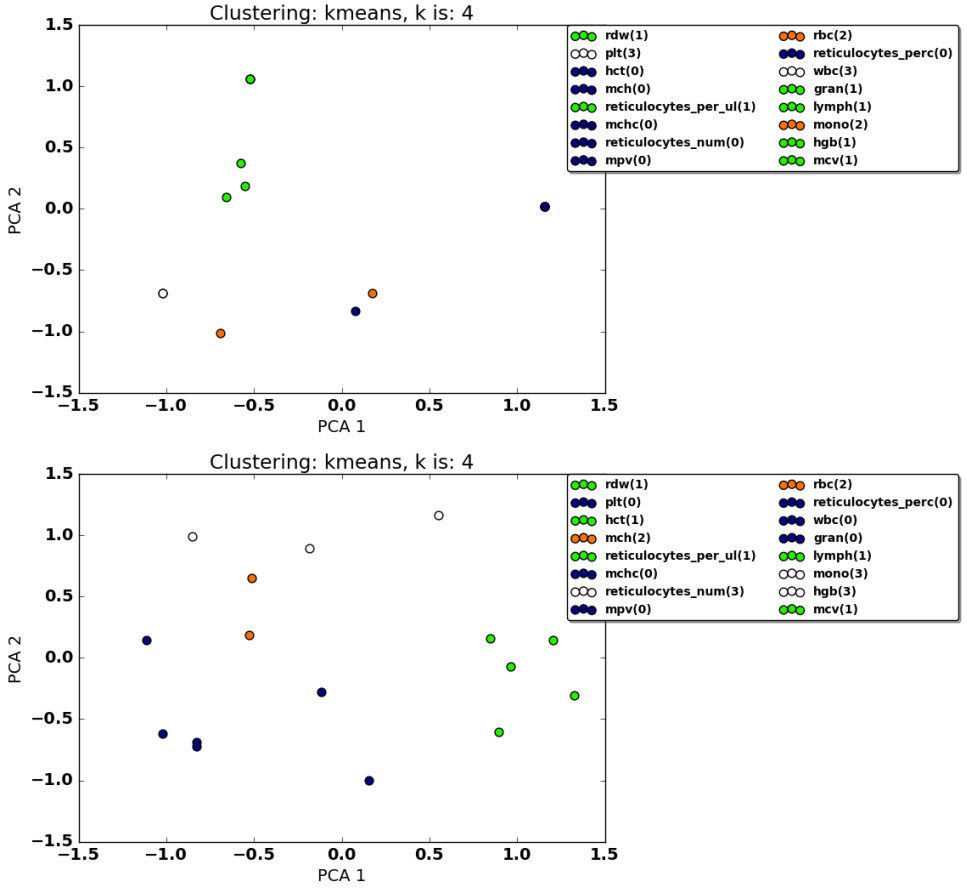


(b) $k = 5$, post-shifting on bottom

FIGURE 4.36: Over all days (omitting RFv13), clustering the match-sign matrices to characterize each clinical parameter.



(a) Silhouette scores, post-shifting on the right



(b) $k = 4$, post-shifting on bottom

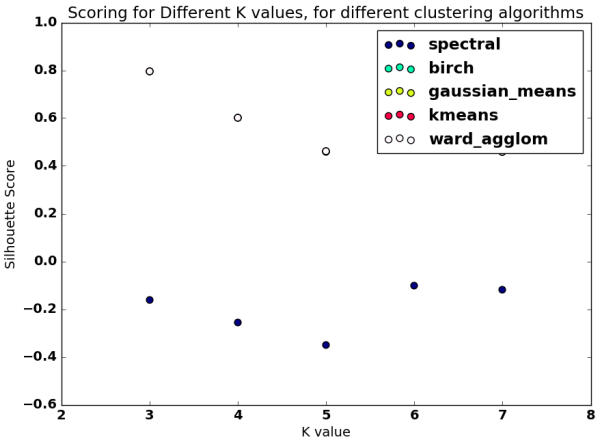
FIGURE 4.37: Up until day 23 (including RFv13), clustering the match-sign matrices to characterize each clinical parameter.

4.3.3 Other Experiments

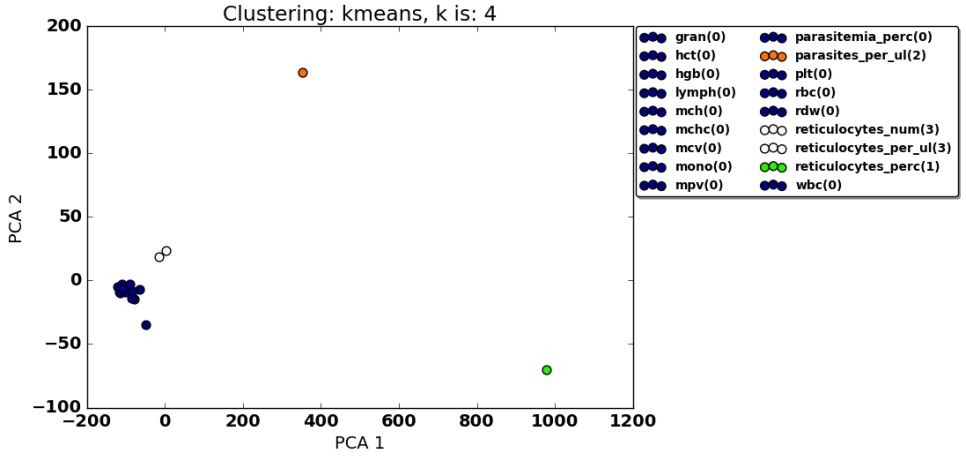
As mentioned before, this framework for analysis can be applied to other experiments. I applied clustering to E03 and E23 as examples, after first following the same steps of Gaussian fitting and residual matrix calculations. While I did not use Bayesian Optimization on these experiments, it still could be implemented; I included these analyses only as proof of concept for the generalization of my thesis methodology to other studies.

E03 is another experiment on rhesus macaques, but with a different species of malaria: *P.coatneyi*. The clinical parameters still cluster similarly; in Figure 4.38 for $k = 5$, all white blood cell types are in one cluster (also including platelets), with red blood cell related parameters in another. Parasite level is again clustered alone. The silhouette score decreases as k increases, but this is likely skewed by the large distance from parasites per microliter and percent reticulocytes.

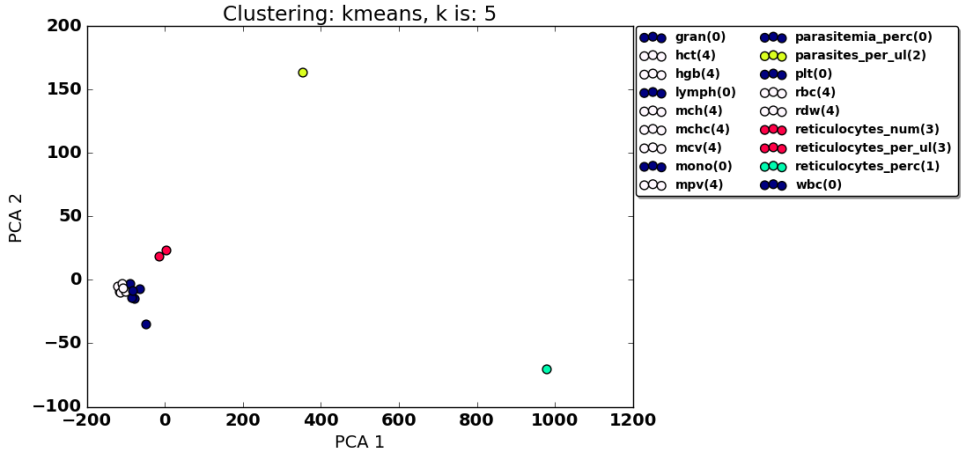
E23 follows a similar course to E04, the main experiment analyzed in this paper, in that rhesus macaques were infected with a different strain of the same parasite species, *P.cynomolgi*. It is therefore unsurprising that the clusters found here are similar to those in Figure 4.34. The clusters in both E04 and E23 also correspond to those in Figure 4.38; the only exception is that white blood cell total count is clustered with reticulocytes. Cluster composition is shifted but still very similar, showing that although the immune response may be species-specific, the integrity of relationships between certain clinical parameters is kept.



(a) Silhouette scores

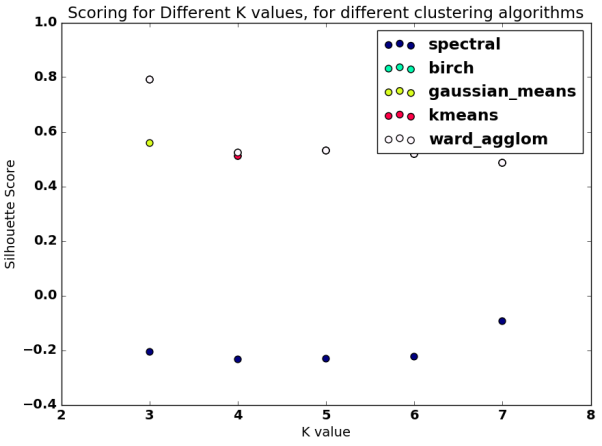


(b) $k = 4$

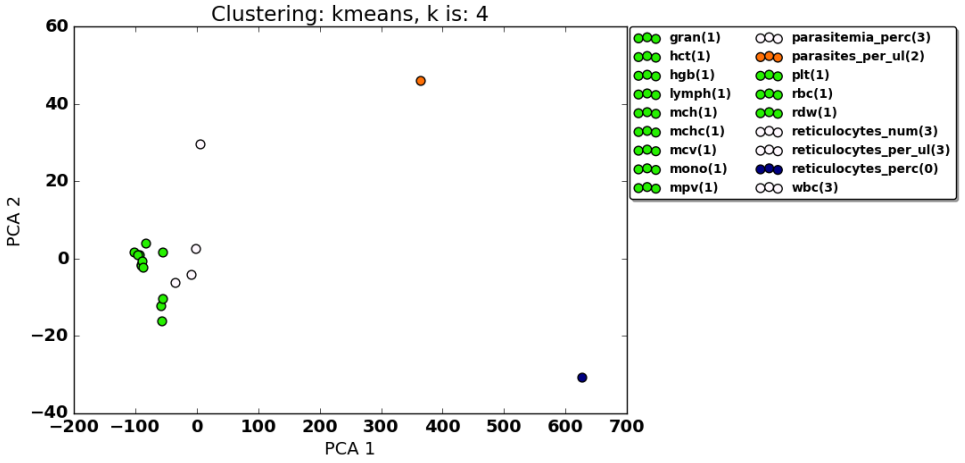


(c) $k = 5$

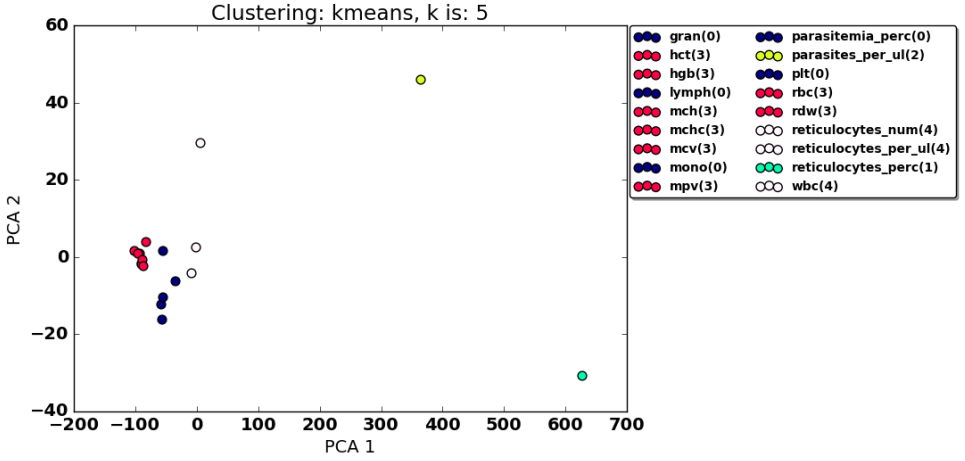
FIGURE 4.38: **E03**: Clustering the residual matrices to characterize each clinical parameter.



(a) Silhouette scores



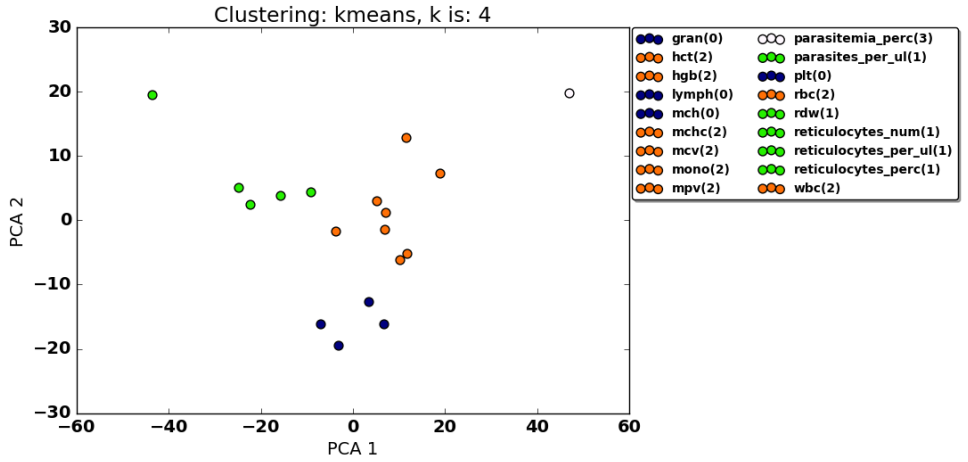
(b) $k = 4$



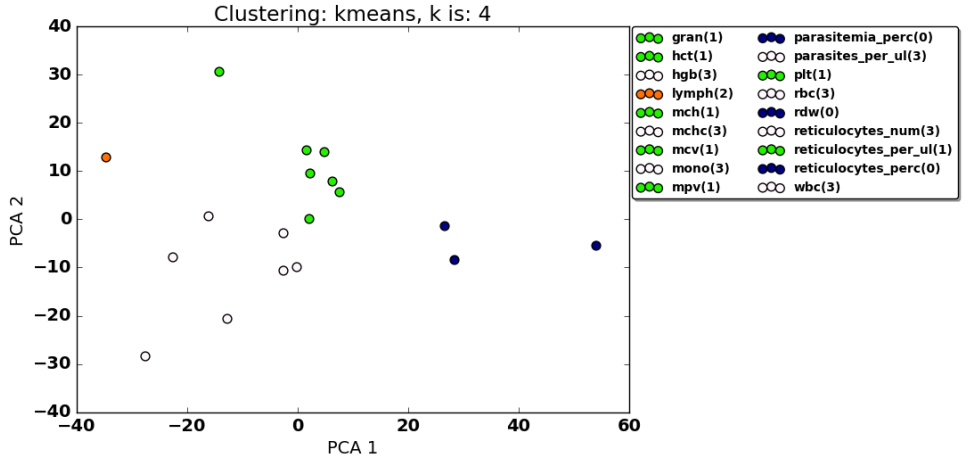
(c) $k = 5$

FIGURE 4.39: **E23**: Clustering the residual matrices to characterize each clinical parameter.

In terms of comparing normalized results for these other two experiments in relation to E04, the cluster composition does shift significantly with respect to each experiment. Regarding parasite level, E23 is more similar to E04 than E03, which as mentioned before is reasonable because the two experiments use the same parasite species. However, E23 is a different strain, which could be the reason for discrepancies; parasite level is clustered with relatively more parameters in E04, as E23 shares only hemoglobin, reticulocyte number, and red blood cell count. E23 also includes white blood cell count and monocytes, which could be indicative of differing immune responses to the separate strains. E03, in contrast, groups parasite level with red blood cell distribution width and reticulocytes. Reticulocytes consequently seem to be significant across different strains and species, marking this clinical parameter for further investigation. Normalized results therefore illuminate variation among the strains and species while also highlighting important similarities, helping to further a more generalized model to consider a more diverse pool of data.



(a) $k = 4$, E03



(b) $k = 4$, E23

FIGURE 4.40: Comparing normalized results for E03 and E23.

Chapter 5

Conclusions

This paper is a proof of concept that automatic analysis can be applied as another layer to manual inspection, while complete replacement of manual work can be a future goal. The main results in the Joyner et al. study discuss the possible importance of reticulocytes as indicators of severity, while comparing signs such as thrombocytopenia and anemia (Joyner et al., 2016). In this study, we have shown that some of these insights can be extracted automatically by applying certain data analysis techniques, which shed light on other parameters as well. As an example from the results, monocytes influential in the adaptive immune response could be important in final disease outcome; additionally, clustering results relate thrombocytopenia and anemia, extracting information about these conditions that was unresolved in the actual study.

While this thesis specifically focused on replicating analysis for experiment E04, the same framework can be applied to other malaria experiments (as discussed in Section 4.3.3), and the ideas are transferable to other experiments in general. Because of the few datasets available, this study was limited by the noise that arises with little information; applying these techniques

to other experiments could help characterize specifics of those studies while also adding to the overall data on which to build a larger and more flexible model for disease severity and stage.

5.1 Contributions

The main contribution of this study is providing a step-by-step framework to begin automatically analyzing small biological datasets, starting with fitting mathematical models and ending with clustering and regression modeling. While some studies have used machine learning to help classify types of malaria in human cases (Andrade et al., 2010), this kind of computational approach to small experimental malaria datasets has not been employed. Therefore, this project provides a preliminary study of which methods might work and what kinds of results should be expected, yielding a stepwise framework for approaching these types of datasets.

5.2 Future Work

As mentioned, this entire framework can be applied to other datasets, as seen in Section 4.3.3; different or similar results could both be meaningful in determining important conclusions regarding the specifics of the experiment analyzed. In addition, there are yet more methods to expand or improve the existing framework. Scalability could be achieved by joining Gaussian fitting windows to reduce shifting computation, which could help in applying the framework to other data. Moreover, the residuals could be normalized against the number of Gaussian functions fitted to the data, which might better characterize differences that are minimized by shifting (e.g. two Gaussians that merge into one because of the shifts). While MongoDB was a

good way to store data, a larger and more flexible database could be created for related experiments available on PlasmoDB to include all raw data and the analyses detailed in this paper; for now, there are separate databases to hold shifting parameters and to hold raw data and related preprocessing- an overarching database containing both raw data and all analyses applied would be helpful in determining next steps for new experiments. Lastly, after running this framework on other datasets, a general model could be constructed for malaria to characterize other features even about the parasite (e.g. the species, relapse or not, etc.).

5.3 Challenges and Learning

Many obstacles arose along the way in my attempts to analyze this dataset, most of them concerned with Bayesian Optimization. Because the Spearmint package is better run with more CPUs, since it involves very complicated computations, I wanted to migrate this process to our remote lab machine. Spearmint also requires MongoDB, and as it was precarious to rely on a remote instance (since it was not under my control), I first logged into a Mongo server on my local machine from the remote machine- this workaround was viable until my IP changed, and so instead I had to deploy an online MongoDB cluster. More problems arose, as the free clusters have both connection and collection limitations, and hence I could only run a few optimization calls at once. Each pairwise comparison takes hours, and so running eighteen data-points for multiple monkeys, each requiring pairwise analysis, was impractical. Additionally, this process was nearly impossible to debug, as often the code would yield no results while also raising no errors. For these reasons, I do not have shifting results for any of the experiments save for E04. With more time, I could have resolved these issues to allow for streamlined automation of the entire process from initial Gaussian fitting to regression modeling and clustering.

Additionally, because the dataset is necessarily small, it is difficult to determine what kinds of analysis are appropriate and valid. In machine learning, datasets usually have the opposite shape; rather than few rows and many columns (i.e. five monkeys and many clinical parameters on each), datasets have fewer columns and many rows (thousands of instances and fewer attributes).

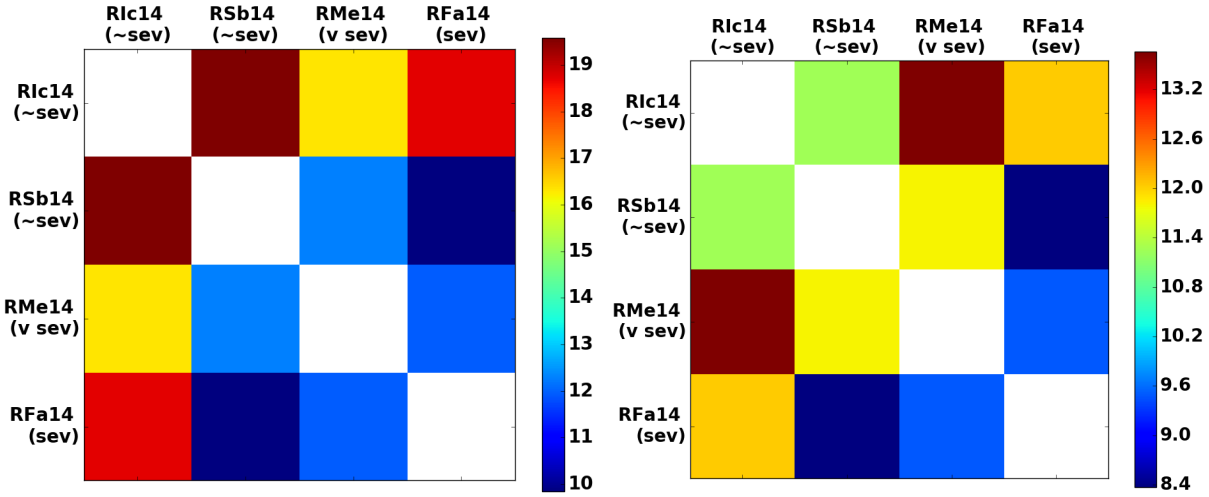
While these challenges impeded analysis considerably, working through them helped me understand how to approach a new dataset; I employed different software packages, ran code remotely, setup and accessed NoSQL databases across various machines, and overall became much better at devising creative solutions. Furthermore, I was using packages for many of the algorithms taught in my Data Mining course, which helped solidify my understanding of the concepts both regarding my thesis and the course. I am confident that I can now tackle novel data via the framework established in this project.

Appendix A

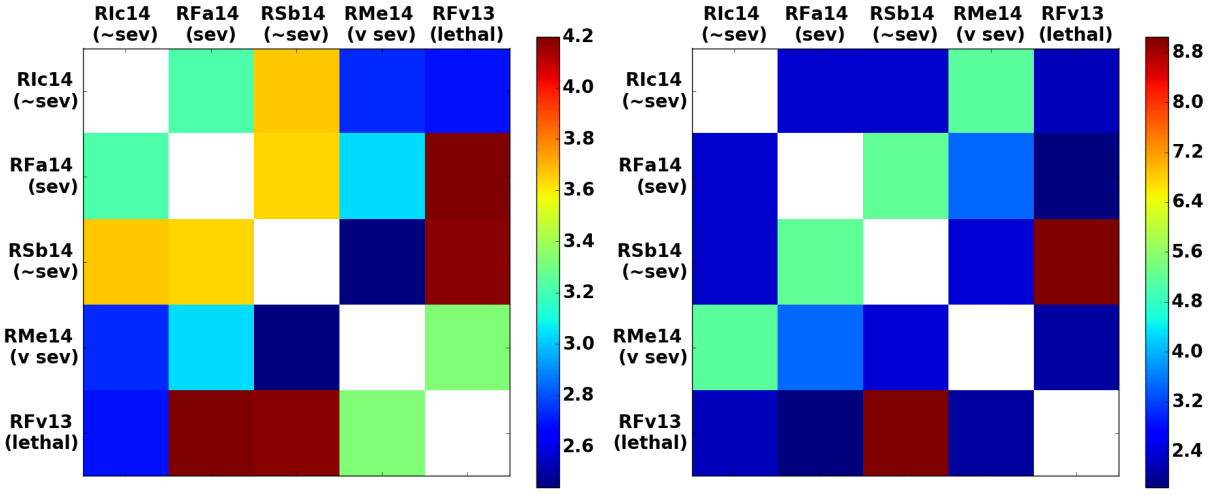
Other Results

A.1 Normalized Results

A.1.1 Residual Matrices

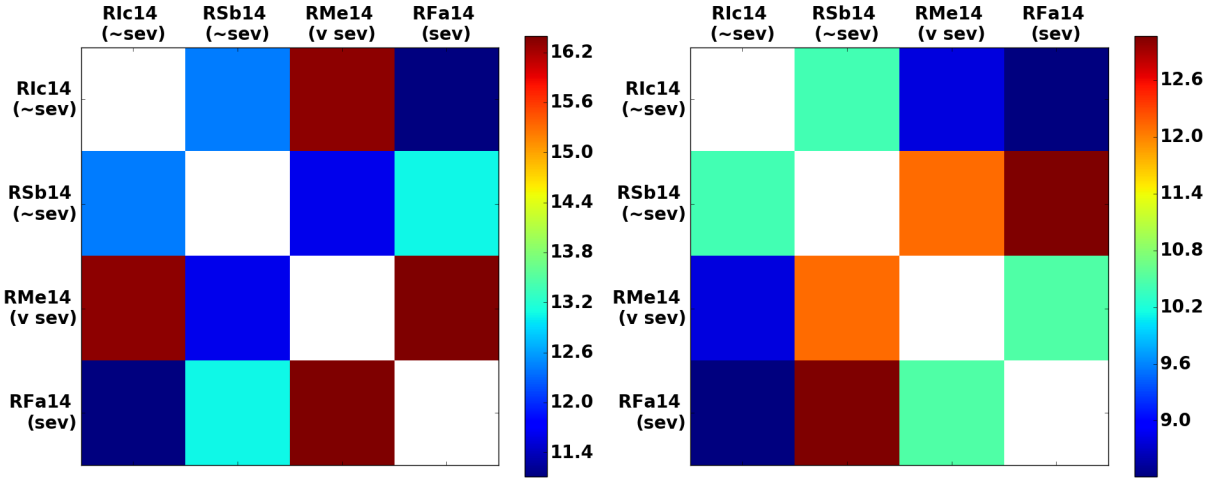


(a) Over all days, post-shifting on the right

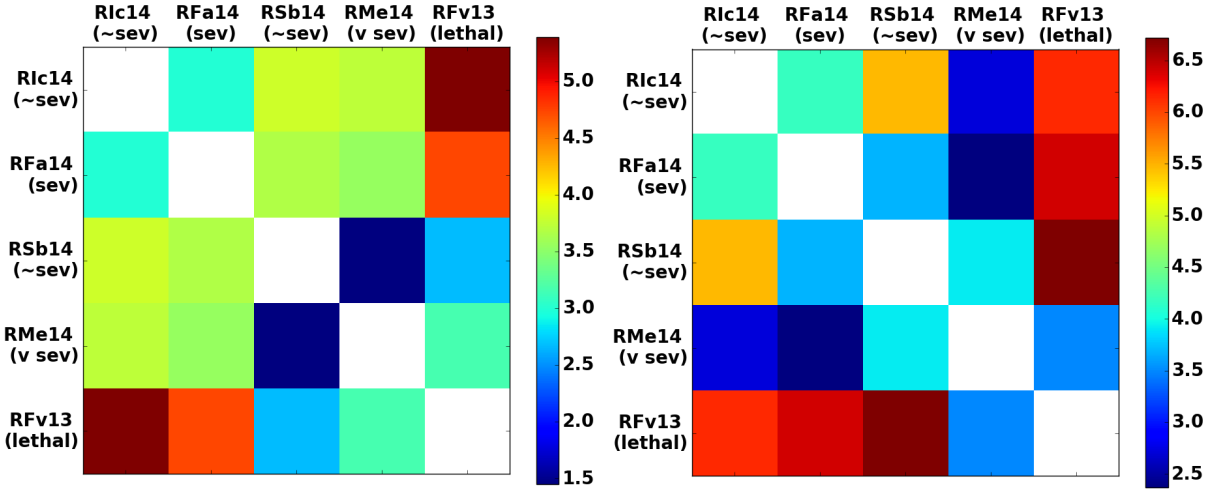


(b) Up to day 23, post-shifting on the right

FIGURE A.1: **gran**: Comparing residual matrices, with and without Bayesian Optimization shifts.

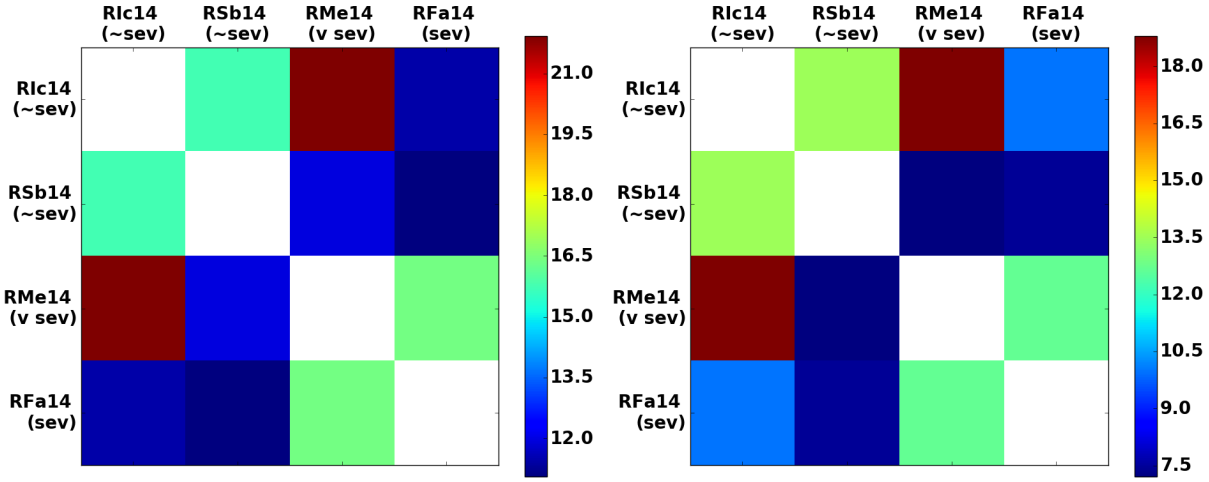


(a) Over all days, post-shifting on the right

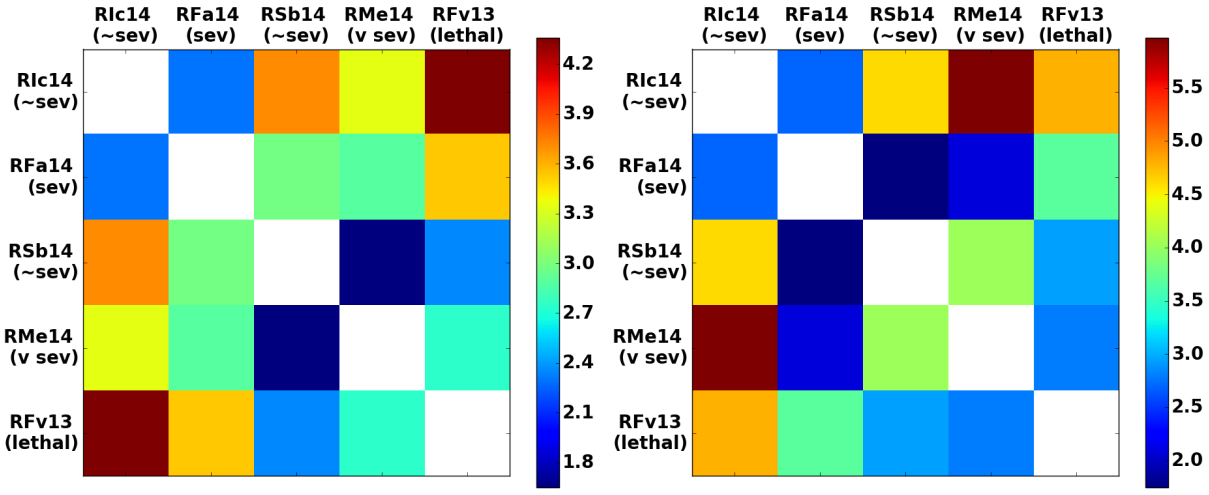


(b) Up to day 23, post-shifting on the right

FIGURE A.2: **hct**: Comparing residual matrices, with and without Bayesian Optimization shifts.

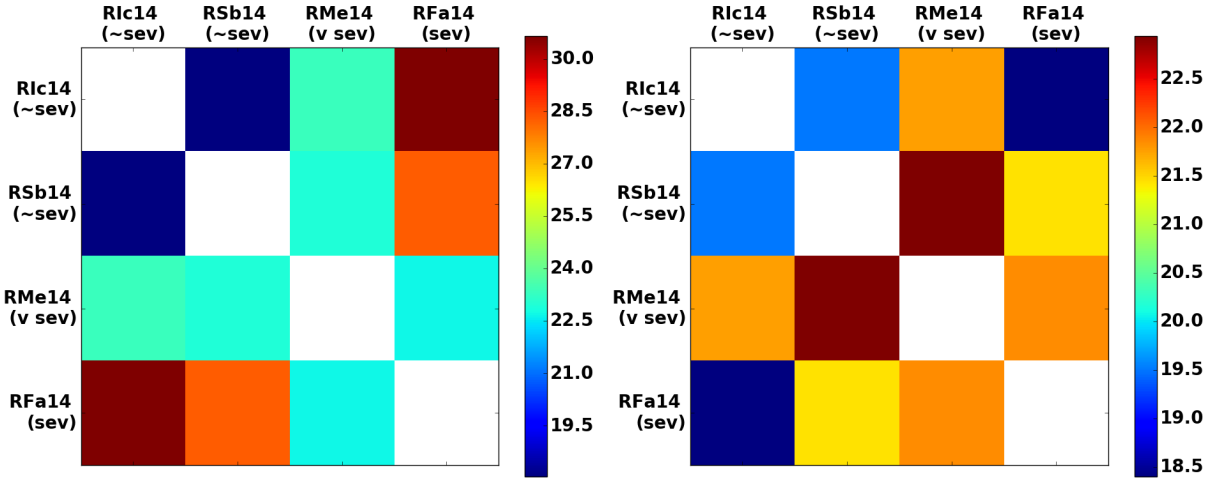


(a) Over all days, post-shifting on the right

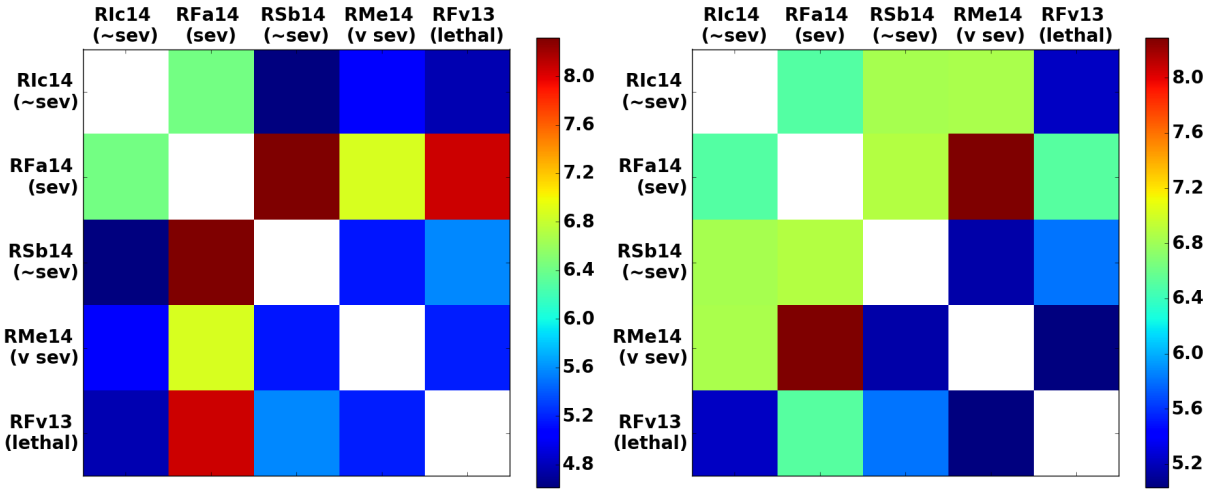


(b) Up to day 23, post-shifting on the right

FIGURE A.3: **hgb**: Comparing residual matrices, with and without Bayesian Optimization shifts.

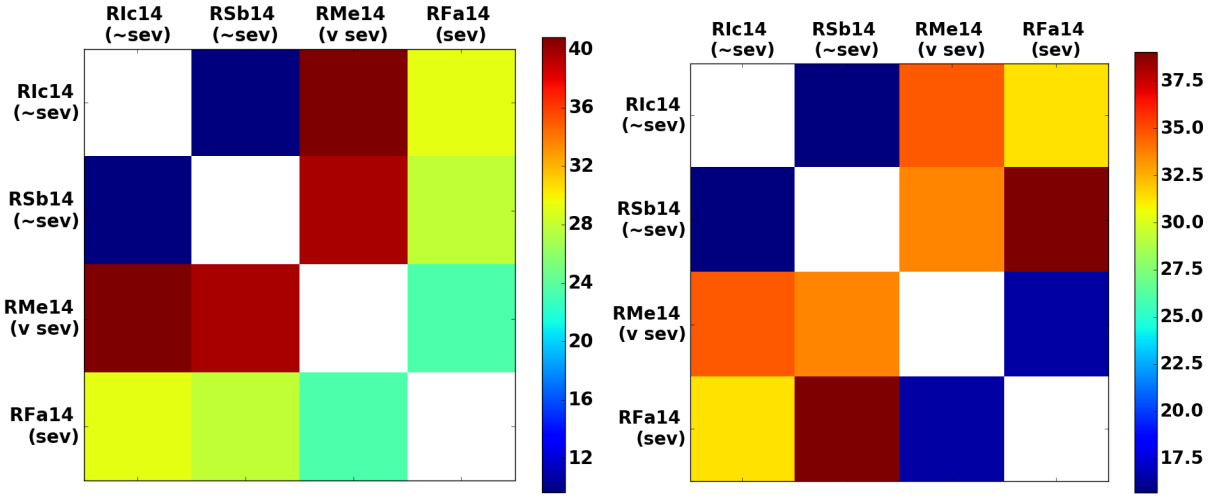


(a) Over all days, post-shifting on the right

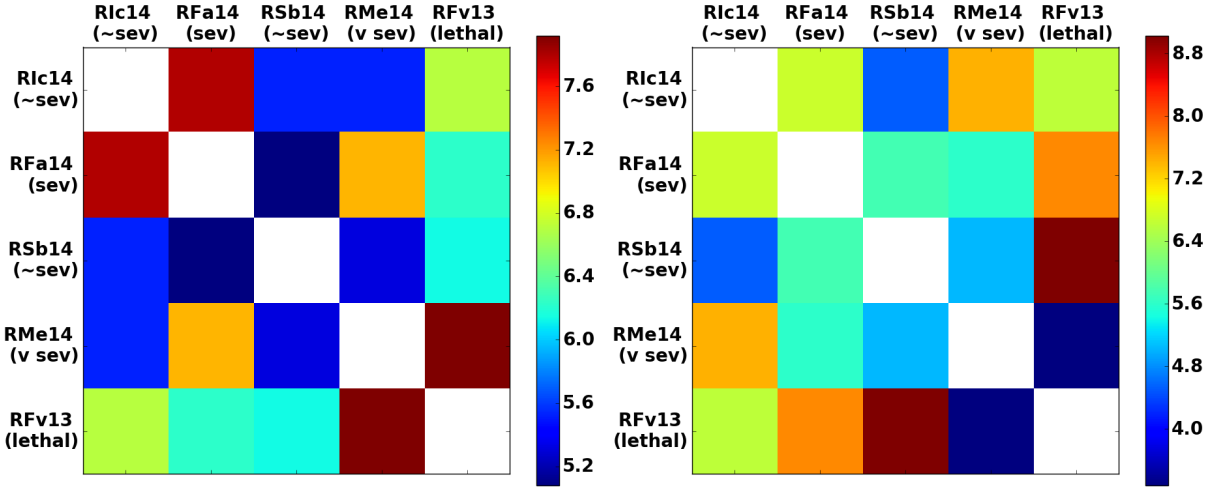


(b) Up to day 23, post-shifting on the right

FIGURE A.4: **lymph**: Comparing residual matrices, with and without Bayesian Optimization shifts.

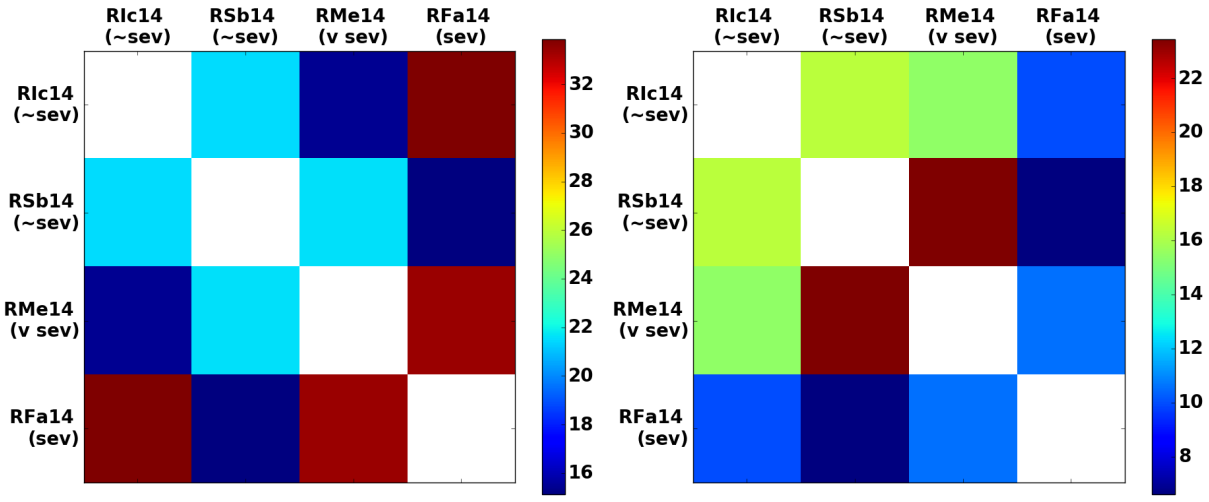


(a) Over all days, post-shifting on the right

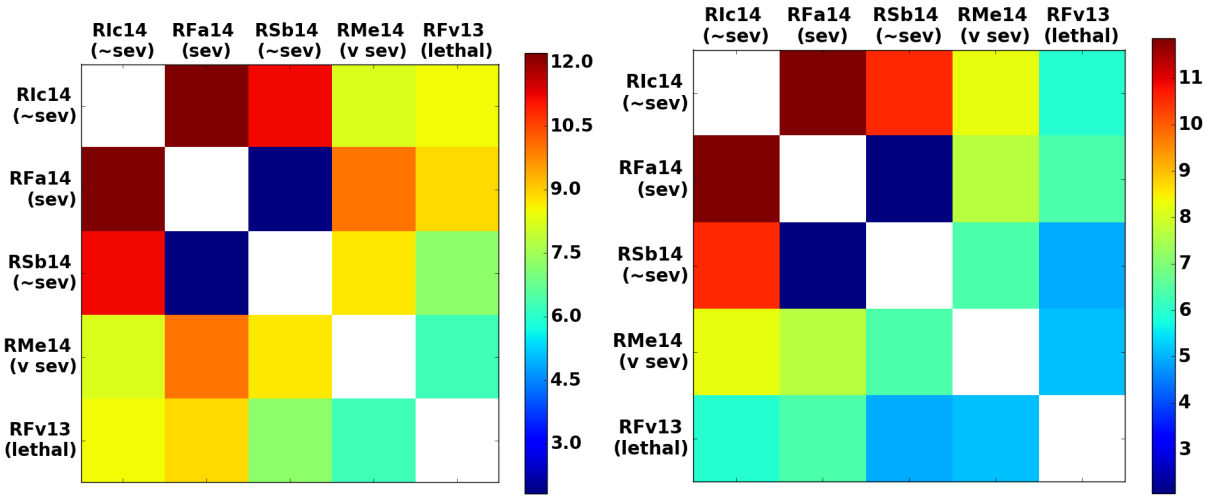


(b) Up to day 23, post-shifting on the right

FIGURE A.5: **mch**: Comparing residual matrices, with and without Bayesian Optimization shifts.

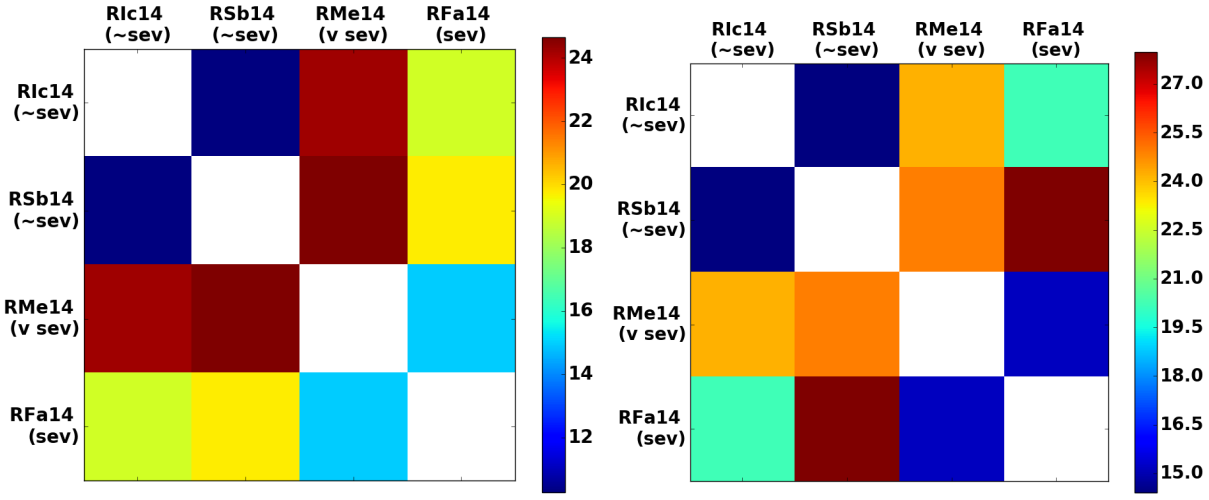


(a) Over all days, post-shifting on the right

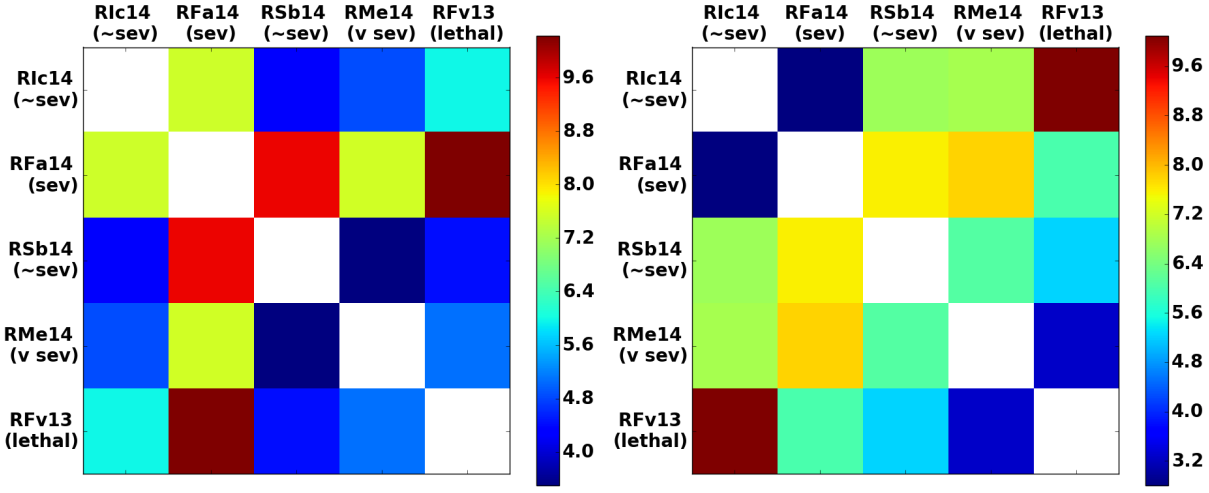


(b) Up to day 23, post-shifting on the right

FIGURE A.6: **mchc**: Comparing residual matrices, with and without Bayesian Optimization shifts.

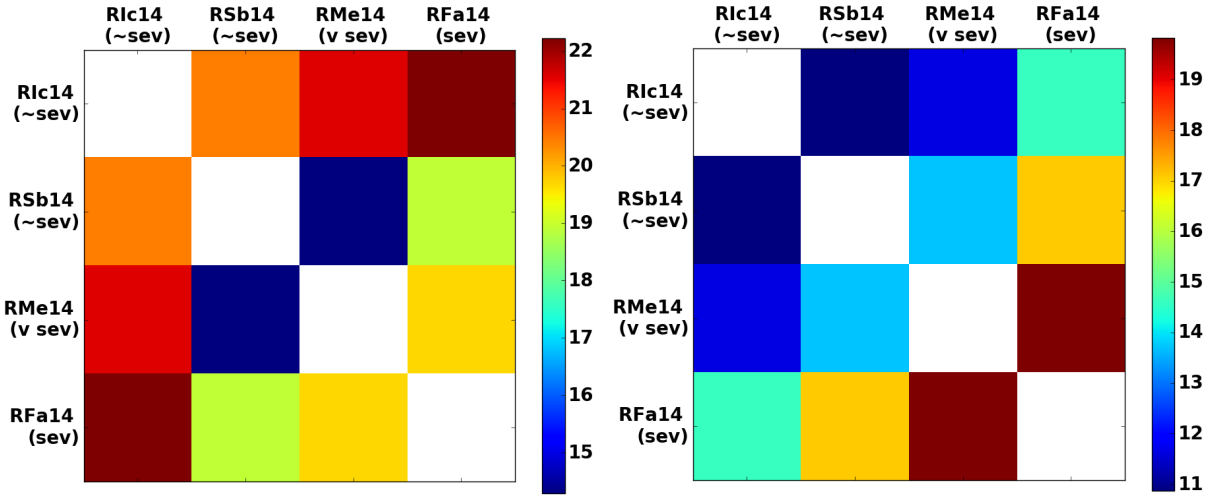


(a) Over all days, post-shifting on the right

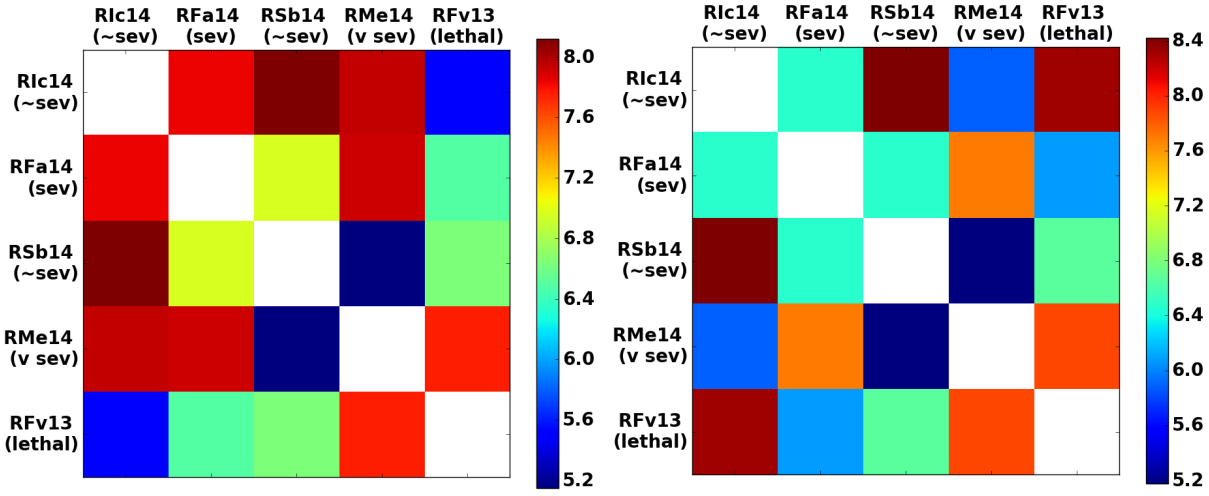


(b) Up to day 23, post-shifting on the right

FIGURE A.7: **mcv**: Comparing residual matrices, with and without Bayesian Optimization shifts.

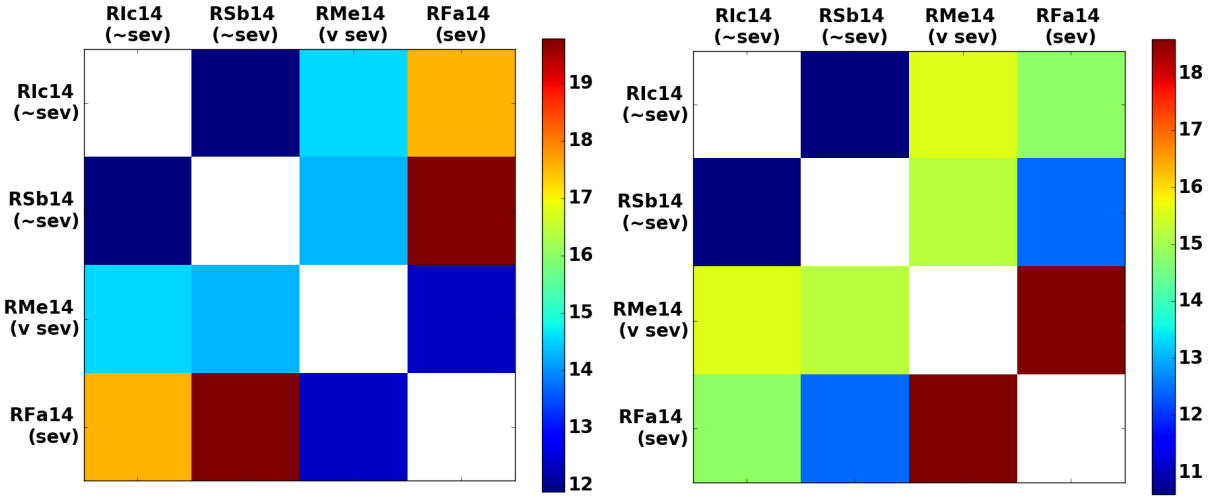


(a) Over all days, post-shifting on the right

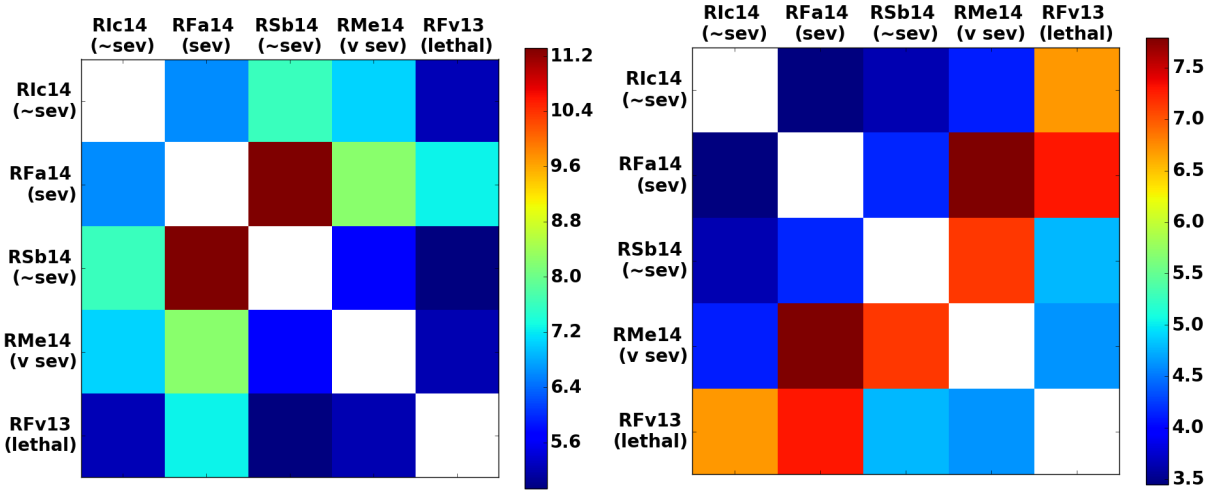


(b) Up to day 23, post-shifting on the right

FIGURE A.8: **mono**: Comparing residual matrices, with and without Bayesian Optimization shifts.

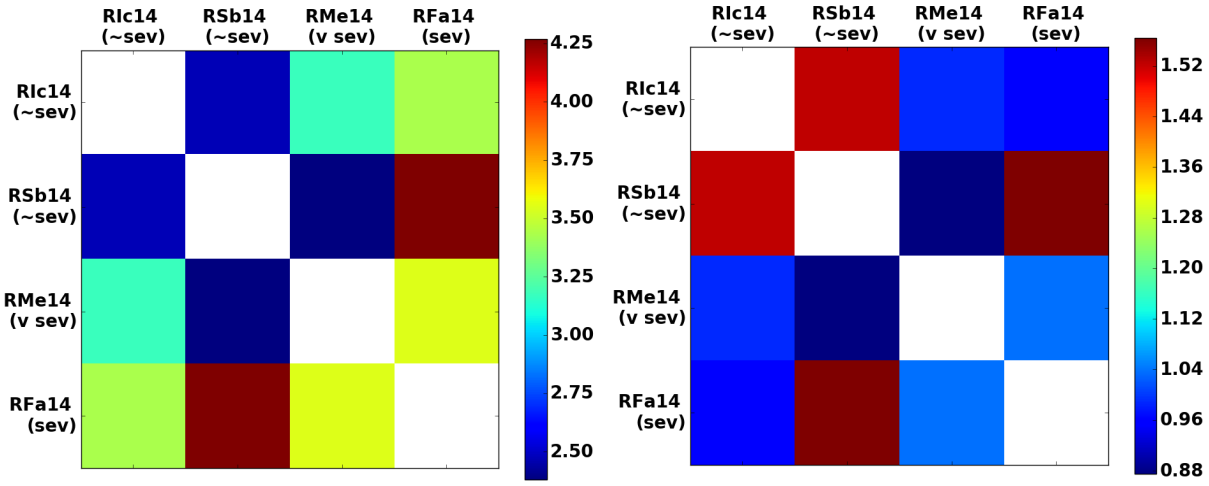


(a) Over all days, post-shifting on the right

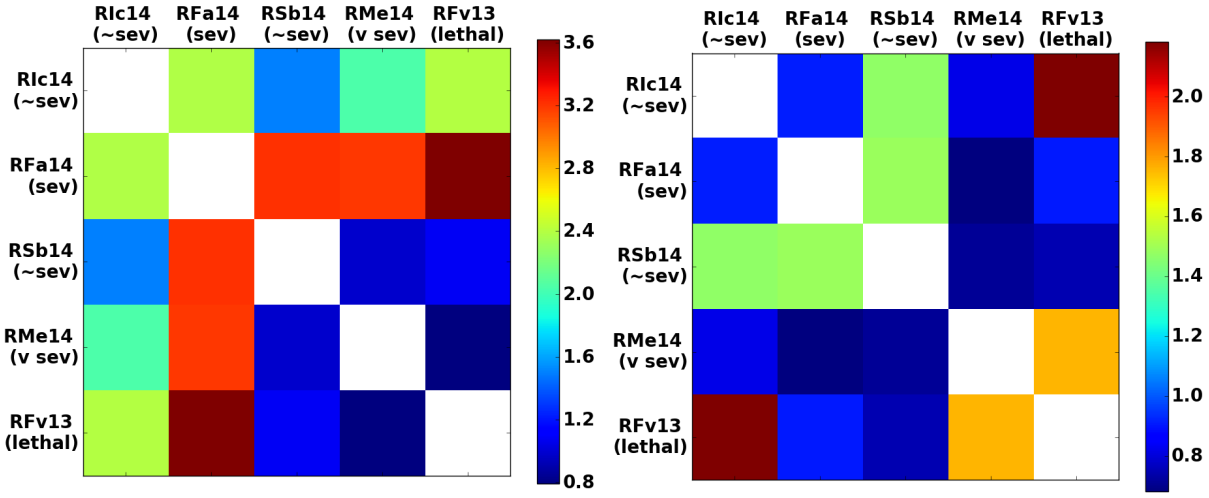


(b) Up to day 23, post-shifting on the right

FIGURE A.9: **mpv**: Comparing residual matrices, with and without Bayesian Optimization shifts.

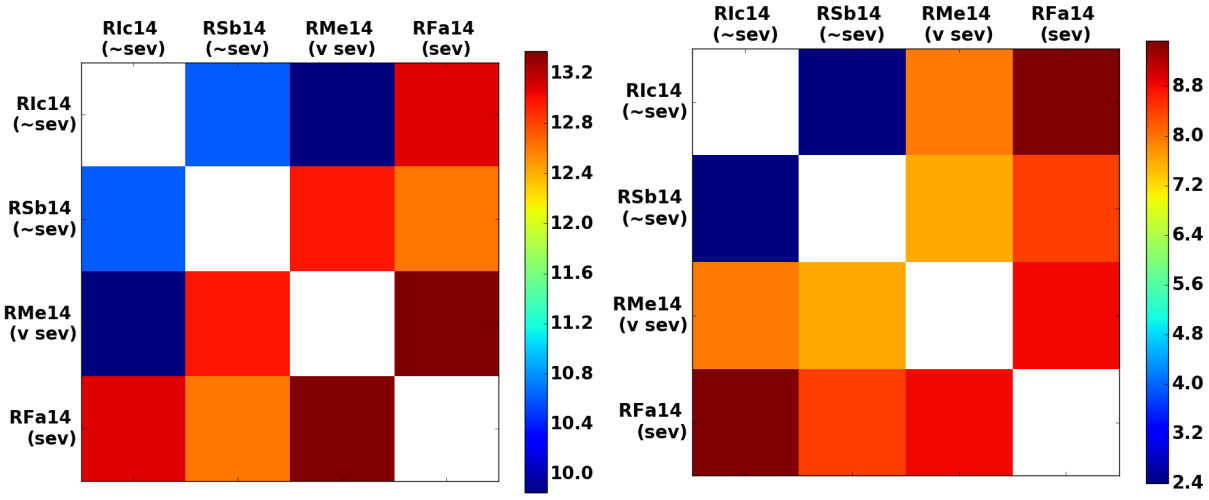


(a) Over all days, post-shifting on the right

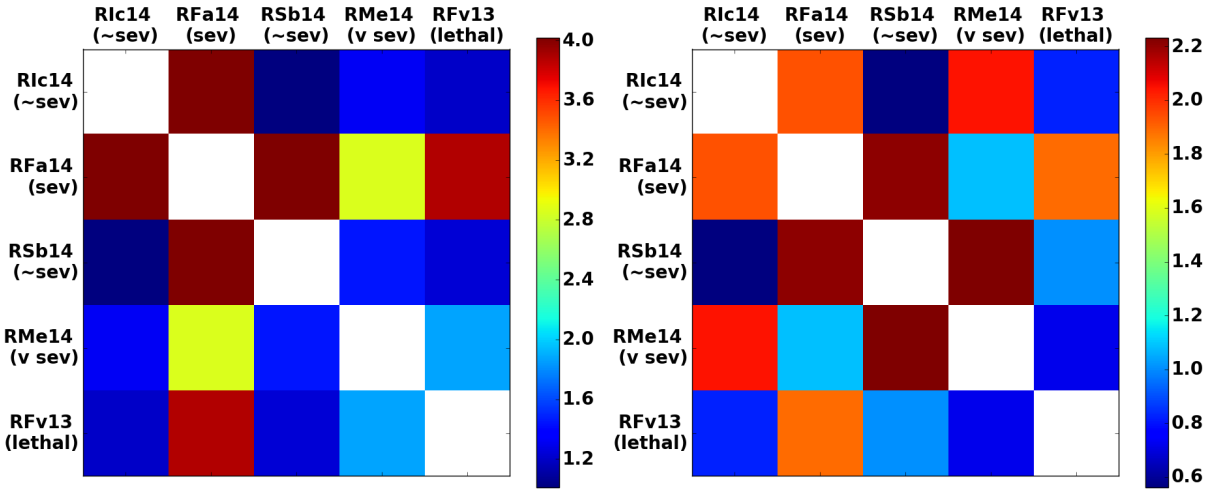


(b) Up to day 23, post-shifting on the right

FIGURE A.10: **% parasitema**: Comparing residual matrices, with and without Bayesian Optimization shifts.

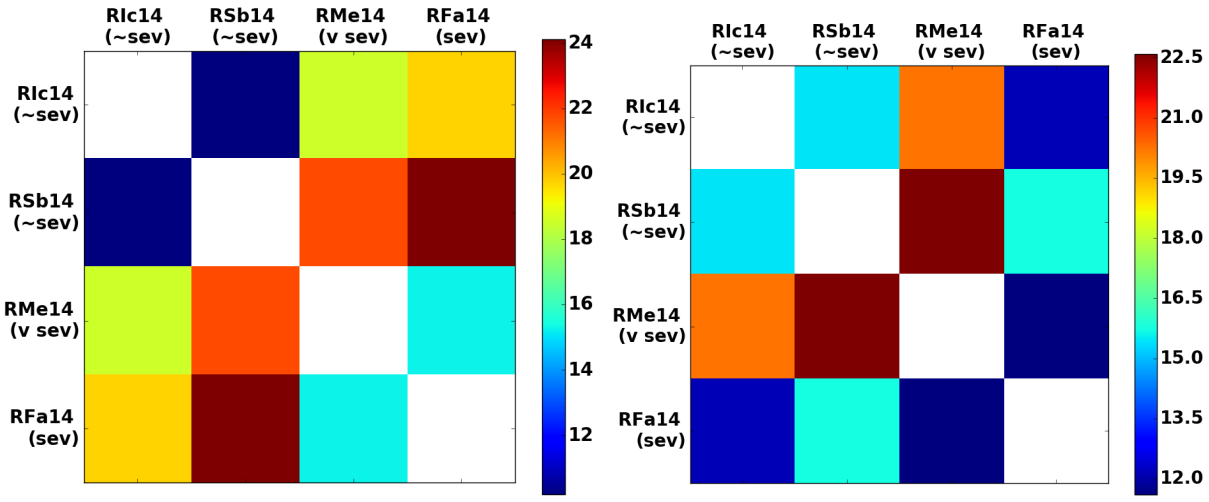


(a) Over all days, post-shifting on the right

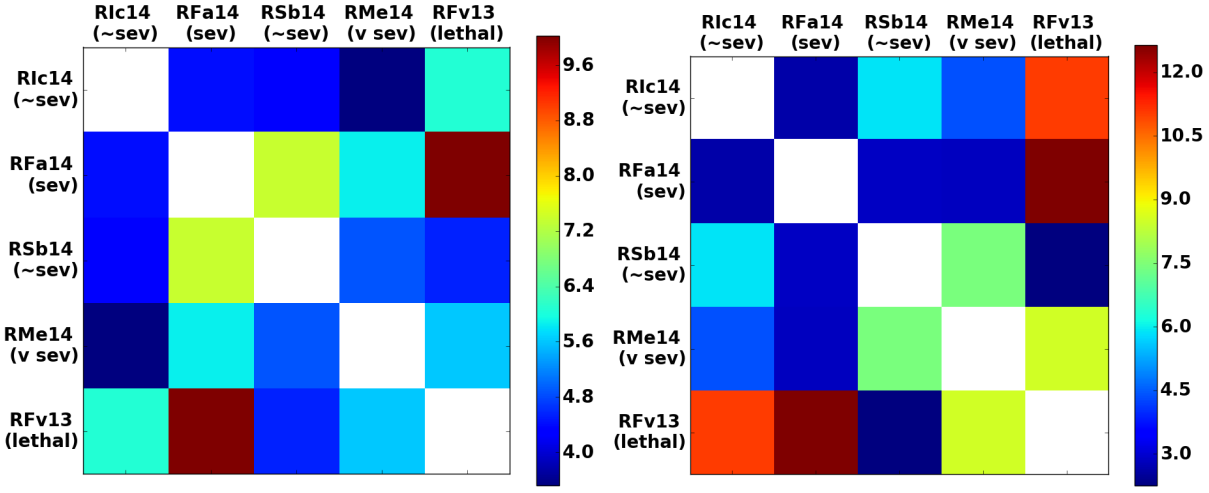


(b) Up to day 23, post-shifting on the right

FIGURE A.11: **parasites** / **uL**: Comparing residual matrices, with and without Bayesian Optimization shifts.

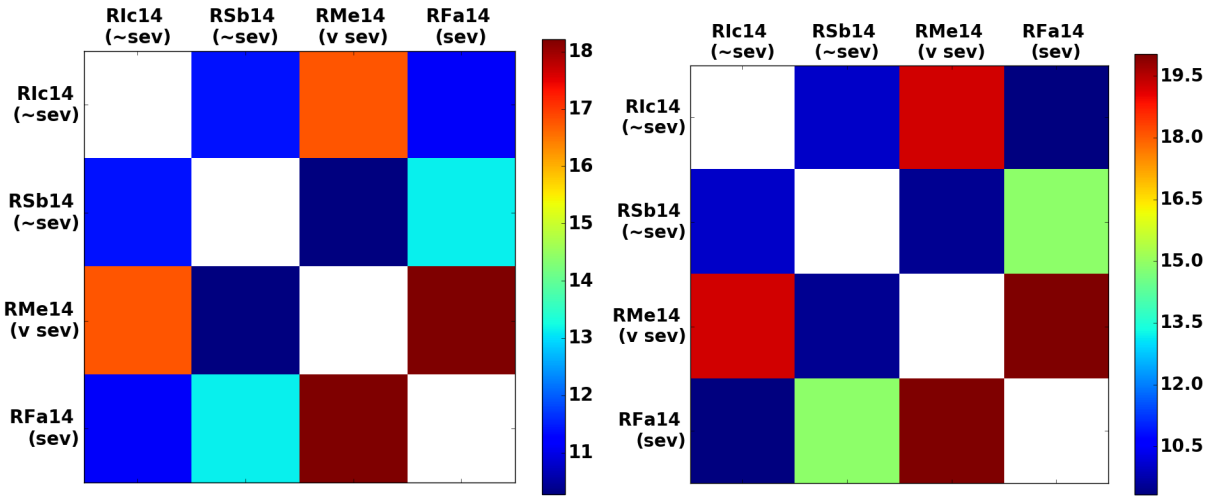


(a) Over all days, post-shifting on the right

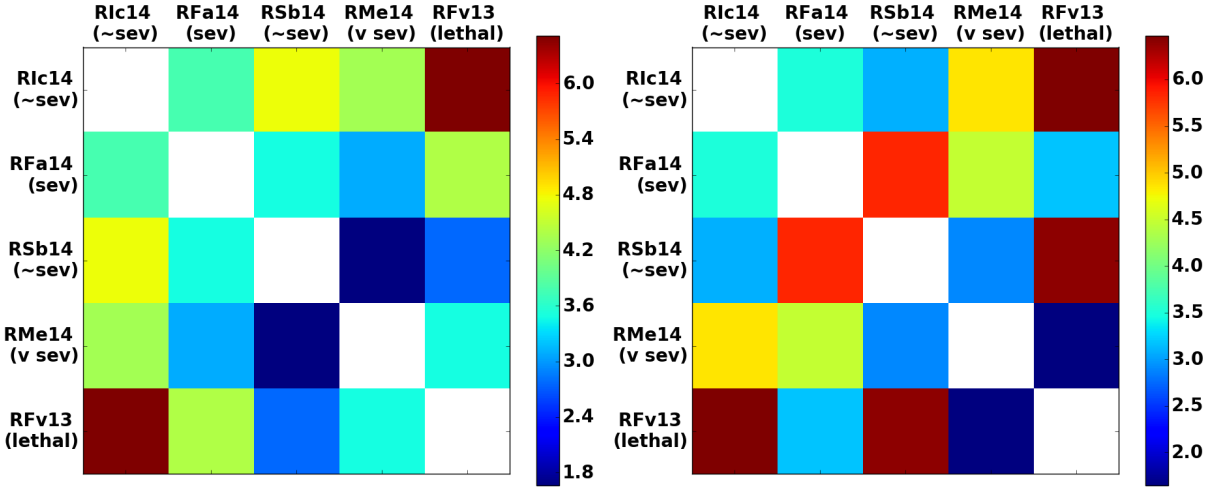


(b) Up to day 23, post-shifting on the right

FIGURE A.12: **plt**: Comparing residual matrices, with and without Bayesian Optimization shifts.

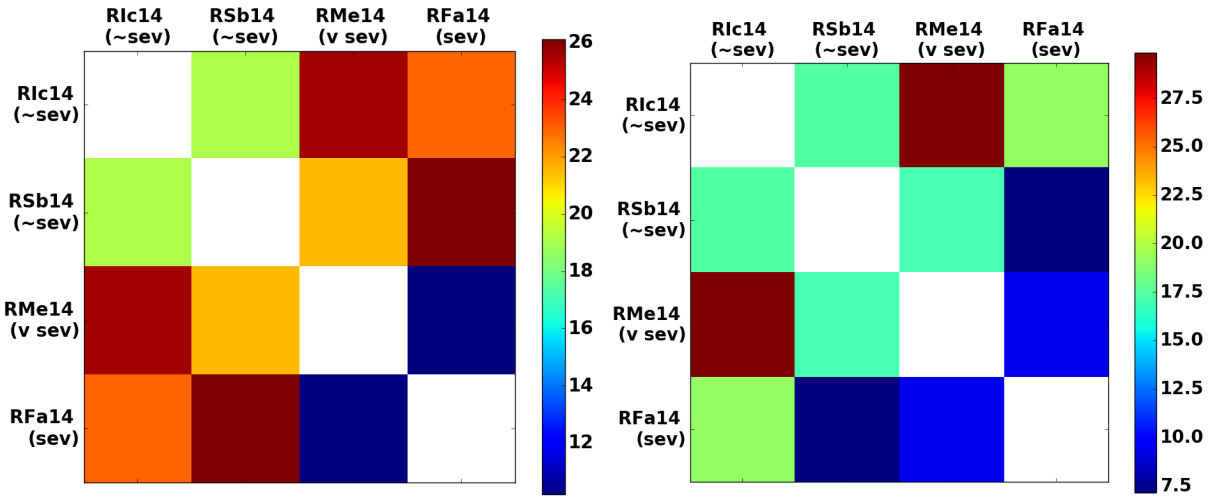


(a) Over all days, post-shifting on the right

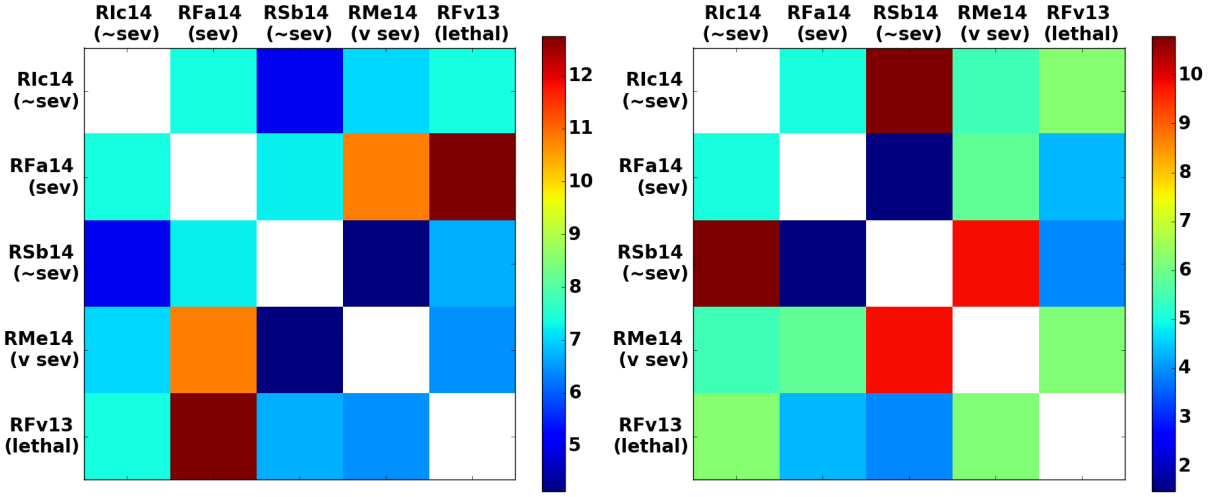


(b) Up to day 23, post-shifting on the right

FIGURE A.13: **rbc**: Comparing residual matrices, with and without Bayesian Optimization shifts.

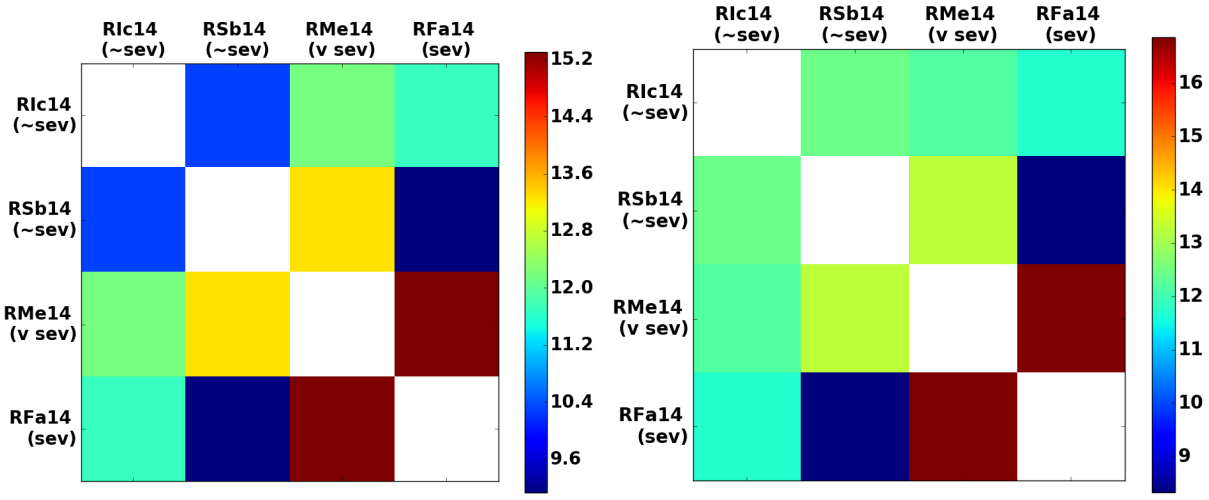


(a) Over all days, post-shifting on the right

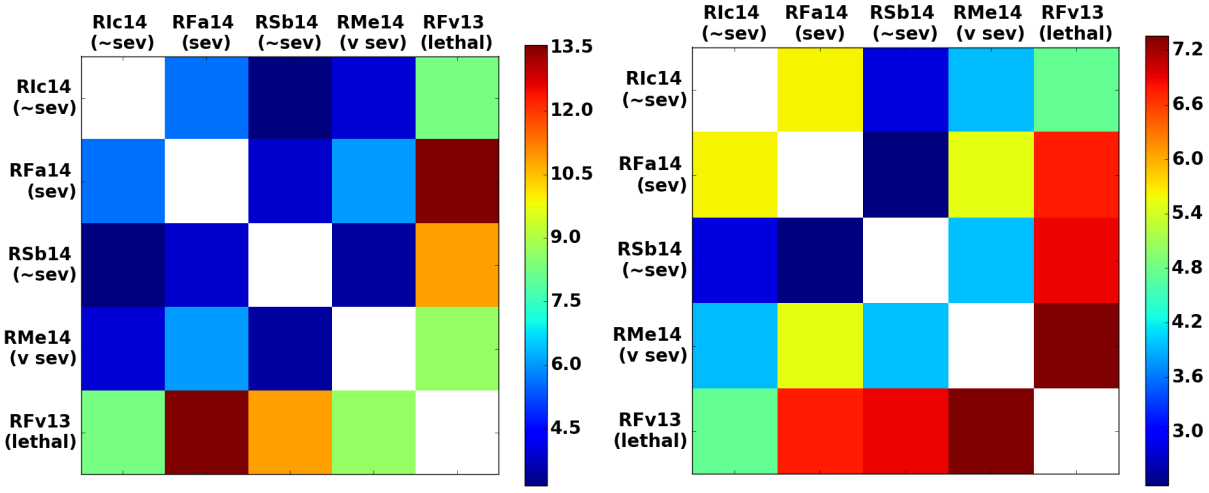


(b) Up to day 23, post-shifting on the right

FIGURE A.14: **rdw**: Comparing residual matrices, with and without Bayesian Optimization shifts.

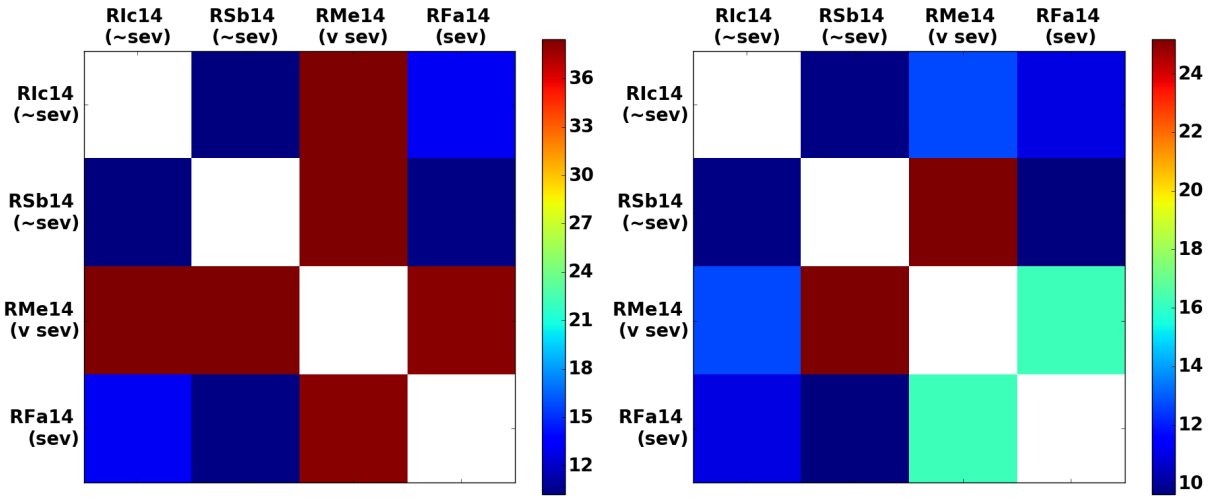


(a) Over all days, post-shifting on the right

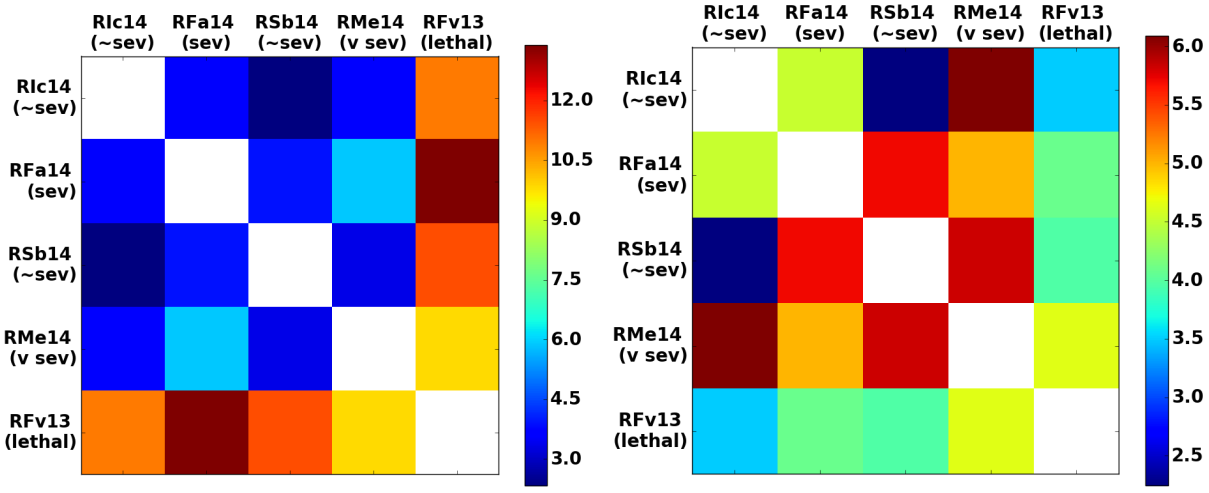


(b) Up to day 23, post-shifting on the right

FIGURE A.15: # **ret**: Comparing residual matrices, with and without Bayesian Optimization shifts.

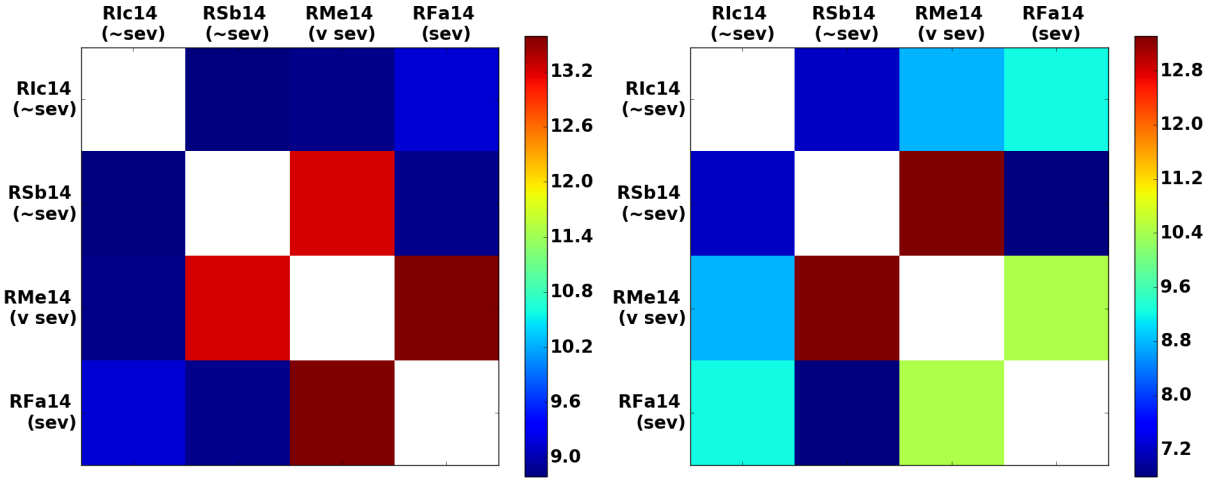


(a) Over all days, post-shifting on the right

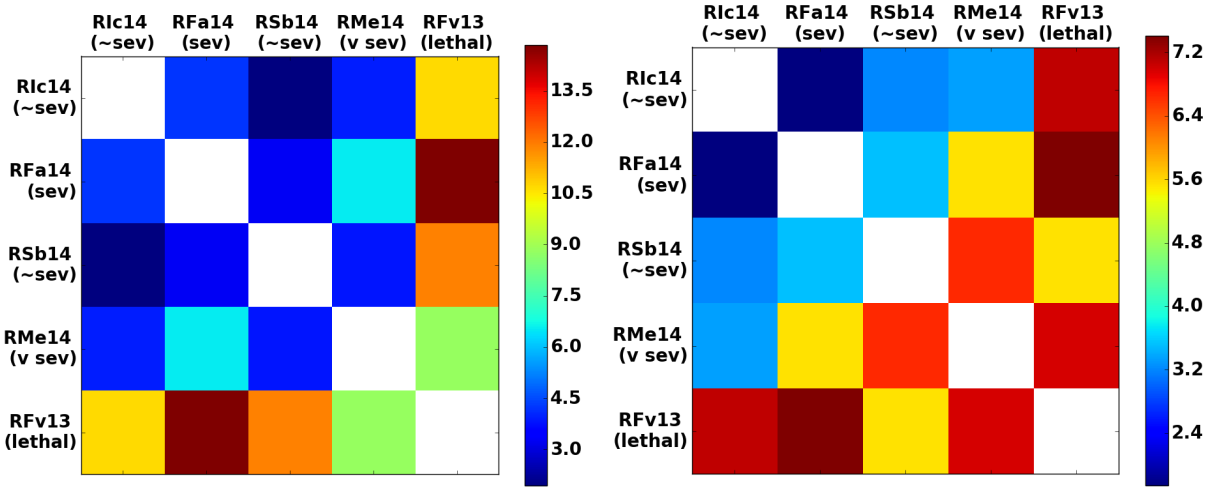


(b) Up to day 23, post-shifting on the right

FIGURE A.16: **reticulocytes / uL**: Comparing residual matrices, with and without Bayesian Optimization shifts.

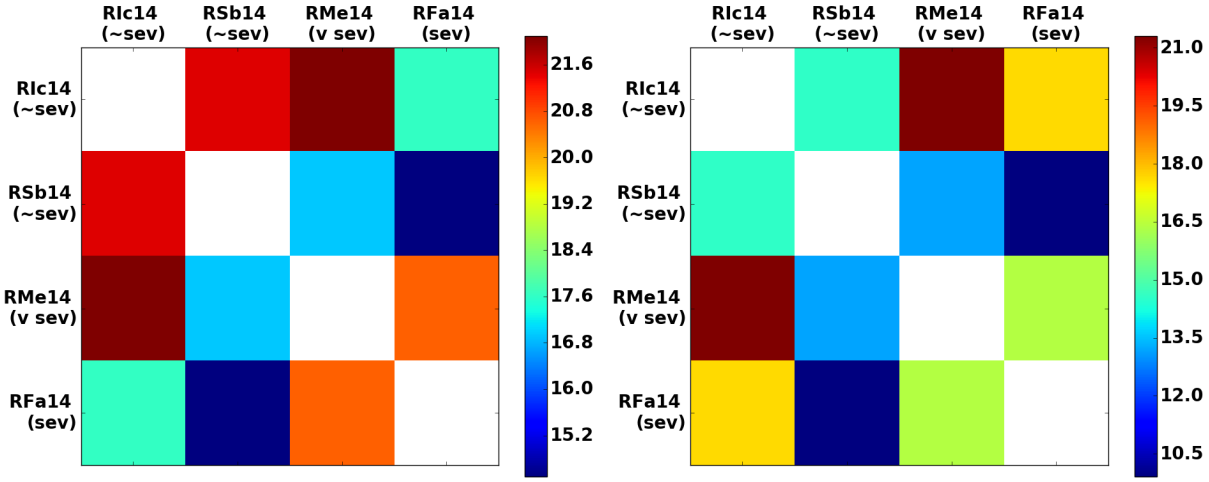


(a) Over all days, post-shifting on the right

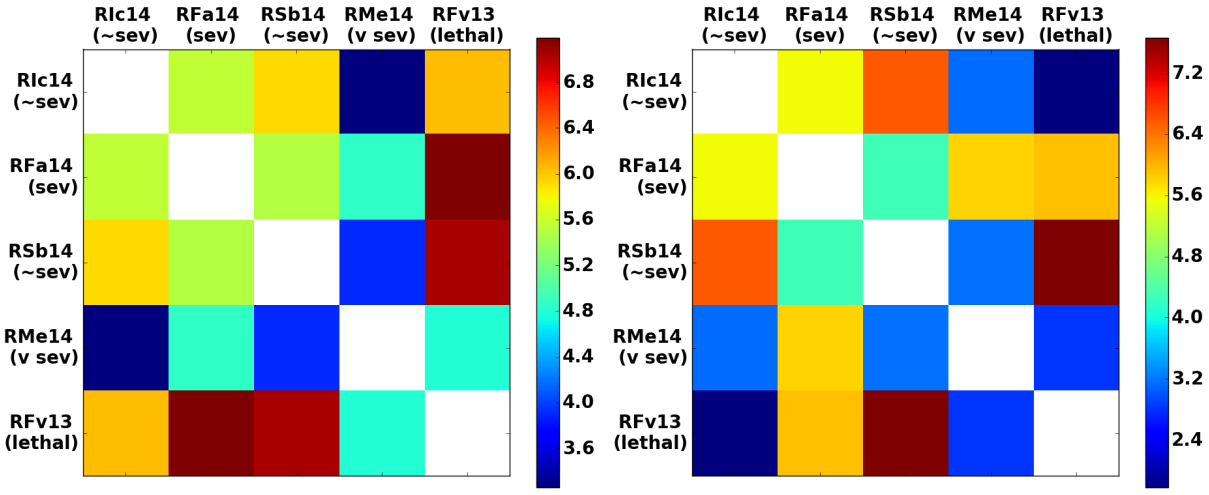


(b) Up to day 23, post-shifting on the right

FIGURE A.17: **% ret**: Comparing residual matrices, with and without Bayesian Optimization shifts.



(a) Over all days, post-shifting on the right



(b) Up to day 23, post-shifting on the right

FIGURE A.18: **wbc**: Comparing residual matrices, with and without Bayesian Optimization shifts.

Appendix B

Abbreviations

Table B.1: Abbreviations for Clinical Parameters

Abbreviation	Description	Units
gran	proportion of granulocytes relative to white blood cell counts (type of white blood cell that has granules which release enzymes during infection)	as a % of white blood cells
hct	red blood cells / total blood volume	as a % of red blood cells to total blood volume
hgb	hemoglobin in blood	grams/deciliter (g/dL)
lymph	proportion of lymphocytes relative to white blood cell counts (B cells / T cells)	as a % of white blood cells
mch	mean corpuscular hemoglobin (hgb in terms of mass / weight)	picograms

mchc	mean corpuscular hemoglobin concentration (hgb per unit volume of red blood cells)	g/dL
mcv	mean corpuscular volume (mean volume of red blood cells)	fL (femtoliters)
mono	proportion of monocytes relative to white blood cell counts (part of adaptive immunity: differentiate into macrophages and dendritic cells)	as a % of white blood cells
mpv	mean platelet volume (average size of platelets)	fL
parasitemia %	percentage of infected red blood cells	(parasites / uL) / (rbc / uL) * 100
parasites / uL	parasitemia % with respect to red blood cell concentration in the complete blood cell count (i.e. total platelets, red blood cells, white blood cells, hemoglobin)	
plt	# platelets / uL	
rbc	# red blood cells / uL	
rdw	red blood cell distribution width (variation in size of rbc's - increased variation denotes illness)	degree of variation, so % deviation
ret	reticulocytes: immature red blood cells (increased numbers show illness, i.e. being made too quickly)	# ret / uL (% ret as % of rbc)
wbc	total white blood cells	white blood cells / uL

References

Ahearn, T. S., Staff, R. T., Redpath, T. W., & Semple, S. I. K. (2005). The use of the levenberg–marquardt curve-fitting algorithm in pharmacokinetic modelling of dce-mri the use of the levenberg–marquardt curve-fitting algorithm in pharmacokinetic modelling of dce-mri. *Physics in Medicine & Biology*, *50*(9), N85-N92.

Aksoy, S., & Haralick, R. (2000). Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters - Special Issue on Image and Video Retrieval*.

Andrade, B. B., Reis-Filho, A., Barros, A. M., Souza-Neto, S. M., Nogueira, L. L., Fukutani, K. F., ... Barral-Netto, M. (2010, May 06). Towards a precise test for malaria diagnosis in the brazilian amazon: comparison among field microscopy, a rapid diagnostic test, nested pcr, and a computational expert system based on artificial neural networks. *Malaria Journal*, *9*(1), 117. Retrieved from https://doi.org/10.1186/1475-2875-9-117 doi: 10.1186/1475-2875-9-117

Baird, J. K., Valecha, N., Duparc, S., White, N. J., & Price, R. N. (2016, March). Diagnosis and treatment of plasmodium vivax malaria. *The American Journal of Tropical Medicine and Hygiene*, *95*(6), 35-51.

CDC. (n.d.). *Malaria.* Retrieved 2017, from https://www.cdc.gov/malaria/

Gavin, H. P. (2011). The levenberg-marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, 1-15.

Gomez, G. (2010, May). *Normalization methods and data preprocessing.* Bioinformatics Unit CNIO.

Joyner, C., Moreno, A., Meyer, E. V., Cabrera-Mora, M., Consortium, T. M., Kissinger, J. C., ... Galinski, M. R. (2016). Plasmodium cynomolgi infections in rhesus macaques display clinical and parasitological features pertinent to modelling vivax malaria pathology and relapse infections. *Malaria Journal*.

Madhulatha, T. S. (2012). An overview on clustering methods. *CoRR*, *abs/1205.1117*. Retrieved from http://arxiv.org/abs/1205.1117

Motulsky, H., & Christopoulos, A. (2004). *Fitting models to biological data using linear and nonlinear regression: a practical guide to curve fitting.* Oxford University Press.

Ng, A. (2004). *Feature selection, l1 vs. l2 regularization, and rotational invariance.* Proceedings of the twenty-first international conference on Machine learning.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Sanger Institute. (n.d.). *Plasmodium cynomolgi.* Retrieved from http://www.sanger.ac.uk/resources/downloads/protozoa/plasmodium-cynomolgi.html

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, *104*(1), 148–175.

Snoek, J., Larochelle, H., & Adams, R. P. (2012, August). *Practical bayesian optimization of machine learning algorithms.*

Stern, A. S., Donoho, D. L., & Hoch, J. C. (2007, October). Nmr data processing using iterative thresholding and minimum l1-norm reconstruction. *Journal of Magnetic Resonance*, *188*(2), 295-300.

WHO. (2017). World malaria report 2017. Retrieved from http://www.who.int/malaria/publications/world-malaria-report-2017/report/en/

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data mining: Practical machine learning tools and techniques* (4th ed.). Eksevuer.