Parameter Fitting

As mentioned in Materials and Methods, to estimate the initial guesses for the parameter estimation for model **1**, we used a hill climbing Monte Carlo algorithm. For that we define an objective function

$$J(\mathrm{x}) = \sqrt{\sum_{i=1}^{n} (\log(V(t_i, \mathrm{x})) - \log(\overline{V}(t_i))^2},$$

where n is the number of data points available, $\mathrm{x} = [\alpha, \mu, \mu_1, \tau, \rho, p, k]^T$ is the vector of parameters to be determined, $V(\mathrm{x}, t_i)$ is the viral load, at time t_i and parameters x, predicted by our model and $\overline{V}(t_i)$ represents the data value at time t_i. For each set of parameters, a numerical solution of the initial value problem, $V(t_i, \mathrm{x})$, is generated via a modification of the Euler algorithm, designed to handle the delays of the model (1). The parameter set that best fits the data occurs when $J(\mathrm{x})$ is minimized over the parameter region.

Initially, we choose the parameters randomly from an uniform distribution on a predetermined interval. We first iterated the Monte Carlo algorithm 12,000 times with the following parameter ranges $\alpha \in (0, 10^{-3})$, $\mu \in (0, 1)$, $\mu_1 \in (0, 1)$, $\tau \in (0, 50)$, $\rho \in (0, 0.5)$, $p \in (0, 500)$, and $k \in (0, 10^{-9})$. From this process we select the set of parameters with smallest $J(\mathrm{x})$. We then repeat the procedure (8,000 more iterations), but the parameters are selected from a range around the current best estimate. That

is, for a given parameter, x, the algorithm will look locally in a window between $x - a \times x$ and $x + a \times x$, where a has an initial value of 0.01. By varying the size of this "window," i.e. the a value, we can both search the parameter space and local regions to minimize $J(\mathrm{x})$, by making very small changes in the parameters. This algorithm also has the advantage of avoiding stagnation in local maxima, because one can increase the window width to find fits better than the local minimum of $J(\mathrm{x})$.

Once we found the vector of parameters

$$\hat{\mathrm{x}} = (\hat{\alpha}, \hat{\mu}, \hat{\mu_1}, \hat{\tau}, \hat{\rho}, \hat{p}, \hat{k})$$

that minimizes the functional J, we used these values as the initial guesses in a standard Levenberg-Marquardt search algorithm (2).

We then calculated 95% C.I. for these estimates by bootstraping the fit residuals (3, 4). Briefely, for each patient, let $\hat{\Upsilon} = \{\hat{\epsilon_1}, \hat{\epsilon_2}, ..., \hat{\epsilon_n}\}$ be the set of residuals defined by

$$\hat{\epsilon_i} = \overline{V}(t_i) - V(t_i, \hat{\mathrm{x}}(t_i)),$$

for all data time points t_i, with $i \in \{1, 2, ..., n\}$. Next, we define $\Upsilon^* = \{\epsilon_1^*, \epsilon_2^*, ..., \epsilon_n^*\}$, where, for each i, ϵ_i^* are values drawn at random with replacement from the set $\hat{\Upsilon}$. We form the sets $V_j^* = \{V_{j_1}^*, V_{j_2}^*, ..., V_{j_n}^*\}$ of independent and identically distributed

pseudo-observations, where

$$V_{j_i}^* = V(t_i, \hat{\mathrm{x}}(t_i)) + \epsilon_i^*.$$

Numerically we generate 200 bootstrap samples $V_1^*, V_2^*, ..., V_{200}^*$, and for each of them we used the Levenberg-Marquardt algorithm above to calculate the correspondent vector of parameters. From these, we directly calculate the 95% quantile confidence interval for each parameter estimate.

For the alternative models shown below, we calculated a set of parameters consistent with the data using the Monte Carlo algorithm described.

Alternative Models

To better understand the predictions provided by the model we examined the sensitivity of our results to some of the basic assumptions of the model. We are interested mainly in two issues: the role of the refractory cell population and the nature of both the cytolytic and noncytolytic immune responses.

In the first alternative model, we remove the R population, and assume that the noncytolytic immune response causes infected cells to return to uninfected cells, (i.e., T_1^* goes to T at rate ρ). The results of fitting this model to the data are shown in SI

Fig 3. As uninfected cells increase the virus resurges giving rise to the prediction of multiple viral peaks, which are not seen in the data.

In the second alternative model we remove the noncytolytic effect(i.e., $\rho = 0$). There is no recovered population as we assume recovery is cytokine dependent. To compensate for the generation of uninfected hepatocytes, the best-fit model predicts a large amount of killing of infected cells, resulting in substantial decreases in total cell numbers. This model is also unable to fit the viral load data, which starts increasing at the end of the first phase of decline, due to generation of through proliferation of new uninfected cells (see SI Fig. 4). This version of the model predicts unrealistic reductions in liver size, of up to 90% (SI Fig. 4) because cytoltyic responses are the main mechanism of viral clearance.

To analyze the immune response in more detail, we built three additional models. In the third alternative model, we removed the cytolytic immune response, by setting μ and μ_1 to zero (SI Fig. 5). This model, with two fewer parameters than the original model, was remarkably consistent with the data. Indeed, comparisons based on the F test demonstrate that this model is statistically equivalent to model **1** for all patients ($P > 0.57$). Given SI Fig 5 consistency with the data, we also explored a model

with just a noncytolytic response, but independent from the effector cell dynamics. That is, the term $\rho E T_1^*$ becomes ρT_1^*. We show these fits in SI Fig. 6. This model generates a monophasic decay in viral load (SI Fig. 6). Further, for patients p1 and p4 the model does not fit the peak viral load nearly as well as model **1**. However, when we take into account that this model has only five parameters, instead of eight, there is no statistically significant improvement in the sum of squared residuals in model **1** compared with this simpler model by an F test ($P > 0.27$).

In another alternative model, we consider the effect of removing both the cytolytic and noncytolytic effects by setting E to be zero. This is meant to mimic experiments in which anti-CD8 monoclonal antibodies were used to remove the $CD8^+$ T cells responsible for the cytolytic and noncytolytic immune responses in HBV-infected chimpanzees (5). The results show that the viral load reaches a set point at the peak of the infection and the clearance does not occur (SI Fig. 7). This is similar to the experimental results in CD8-depleted chimpanzees (5).

In the last alternative model we consider two populations of productively infected hepatocytes (T_1^*, T_2^*, with single or multiple cccDNA nuclear copies, respectively). Both types of infected cells can be killed by the immune response at the rate of μE per cell. Both classes of infected cells proliferate in a manner similar to the uninfected cells. However, because cccDNA does not replicate upon cell division, when a

T_1^* cell divides, it will produce one infected cell with one copy of cccDNA, keeping T_1^* unchanged, and one cell with no cccDNA (6), which we put into the T population. Division of infected cells occurs at a growth rate r. Finally, T_1^* can also be lost due to synthesis of new cccDNA and transition into the T_2^* class at rate z. The results are similar with the ones in model **1** (SI Fig. 9). Although we consider this model to be more biological, the results show no statistically significant improvement in the sum of squared residuals in this model compared with model **1** by an F test ($P > 0.92$).

ALT Compared with Rate of Cell Killing

Because ALT is released when a hepatocyte is killed, we compared ALT with the level of effector cells. High levels of effector cells will generate more killing and hence more ALT. Alternatively, one can compare the level of ALT with the rate of killing ($\mu ET^* + \mu_1 ER$). In SI Fig. 8 we make this comparison. Notice that when the rate of killing rises so does ALT. However, when the rate of killing stabilizes, usually a few orders of magnitude below its peak, the value of ALT decreases. Naturally, ALT is also being cleared and this comparison neglects this feature. We have not modeled ALT as this would introduce at least three new parameters (7).

References

1. Ciupe SM, de Bivort B, Bortz D, Nelson PW (2006) *Math Biosci* 200: 1–27.

2. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2002) *Numerical Recipes in C++: The art of scientific computing* (Cambridge University Press).

3. Efron B, Tibshirani RJ (1993) *An introduction to the Bootstrap* (Chapmann and Hall, New York).

4. Davison AC, Hinkley DV (2003) *Bootstarp methods and their applications* (Cambridge University Press, Cambridge).

5. Thimme R, Wieland S, Steiger C, Ghrayeb J, Reimann KA, Purcell RH, Chisari FV (2003) *J Virol* 77: 68–76.

6. Zhang YY, Zhang BH, Theele D, Litwin S, Toll E, Summers J (2003) *Proc Natl Acad Sci USA* 100: 12372–12377.

7. Ribeiro RM, Almer JL, Powers KA, Layden TJ, Perelson AS (2003) *Hepatology* 38: 509–517.