Published in final edited form as:

Curr Protoc Immunol. ; 122(1): 54.

Statistics for Immunologists

Jeff Skinner

5625 Fishers Lane, Room 4N07C, Rockville, MD 20852

Abstract

Statistical tests and graphs are an important part of any immunological research publication or presentation, but not every immunologist has the statistical expertise to choose the best methods to evaluate and represent their experiments. These protocols provide a brief overview of the statistical methods most relevant to immunology researchers, so they can either analyze their own data or better understand their statistician collaborators. Protocols cover the basics of t-tests, ANOVA, nonparametric tests, linear and nonlinear regression models, categorical response models, statistical design of experiments and advanced methods like re-sampling tests, Bayesian statistics and methods for high throughput -omics data. Each topic receives a brief overview of the theory and usage, followed by brief instructions and code examples for R software. These protocols should be useful to biologists with any amount of statistical knowledge.

Keywords

statistics; immunology; graphs; parametric; nonparametric; regression; nonlinear; generalized linear model; survival; design of experiments; resampling; Bayesian; high throughput; data mining; cross-validation; R project

INTRODUCTION

The disciplines of biology and statistics have a long history together, and all biologists need a basic understanding of statistical methods to better understand the literature in their field and publish their own new findings. These protocols are designed to provide a basic review of statistical techniques important to the work of contemporary immunologists. They will

(301) 451-4973, skinnerj@niaid.nih.gov.

KEY REFERENCES

There are thousands of statistics textbooks used on college campuses around the world, but here are a few of my favorites that you might find useful. You certainly don't need to buy any of these texts, but each book covers a slightly different type of content if you want a general statistics resource or if you want to learn more about a specific topic. For the graduate-level textbooks, you may want to review textbooks on linear algebra and calculus as well.

INTERNET RESOURCES

<https://www.r-project.org/> Primary website describing the R Project for Statistical Computing, developers of the R statistics software used throughout this manual.

<https://www.graphpad.com/scientific-software/prism/> Website for Graph Pad Prism, a commercial software was for graphing and statistical analyses of simple biological experiments

<http://www.jmp.com/> Website for JMP Statistical Discovery, a commercial statistics and design of experiments software from SAS Institute Inc.

<http://www.itl.nist.gov/div898/handbook/index.htm> This is the Engineering Statistics Handbook, produced by the U.S. National Institute of Standards and Technology (NIST). It's a great statistics reference, but Chapter 5 "Improve", in particular, is a great reference for DOE.

cover a wide variety of statistical topics with some basic explanations and examples to guide you towards some reasonable methods for different immunology experiments. Of course, these protocols cannot serve as a total substitute for proper statistics course work or statistics text books, but hopefully they can serve as a useful primer for any statistical problems you might encounter from the immunology bench.

STRATEGIC PLANNING

A flow chart (Fig. 1) describes the organization of the statistical concepts presented here, starting with statistical graphs and branching out into different statistical tests and concepts for specific types of immunological data. One key idea is the distinction between predictor variables (i.e. independent variables), which are typically variables we can control or variables we expect to influence an outcome, and response variables (i.e. dependent variables), which are typically the variables outside of our control or the variables most influenced by our experimental conditions. Another important concept is the difference between continuous variables (e.g. age, height, weight, mass, concentration of specific compounds, ...) that can be measured precisely in fractions of a unit (e.g. 68.3 inches, 154.8 lbs, 17.1 micrograms, ...), and categorical variables, which have a finite list of discrete outcomes (e.g. gender is male or female, a person's disease status can be infected or uninfected). These concepts and more would be used to determine how a data set should be graphed, which statistical tests should be used to analyze the data and how that information might be used to plan better experiments in the future.

Before delving into specific statistical methods, it may be wise to consider the statistical software you might use to produce your graphs and calculations. There are many statistics software options, some free and others quite expensive, some very simple and others quite difficult to learn. Some statistics software offers a point-and-click graphic user interface (GUI) familiar to most anyone using Apple Macintosh or Microsoft Windows operating systems and their web browsers, office productivity suites, etc. Other statistics software requires the use of typed commands and looks a little bit more like computer programming.

These protocols will mostly use software from The R Project for Statistical Computing (<https://www.r-project.org/>), which is a free and open source software platform that mostly uses typed commands. One key aspect of the R Project software is that its features are broken up into thousands of different "package libraries", most of which need to be downloaded, installed and loaded to use. It sounds inconvenient, but the modular nature of the R Project software forces you to customize the software with your own unique blend of basic features and cutting-edge developments created by statistics and biology researchers from all around the world. The R Project software, R, can be used many ways, but the simplest choice is to download the software and its standard graphic user interface (GUI) from <https://cran.r-project.org>. There are installation instructions available for Windows, Mac and Linux operating systems on the linked webpage. Package libraries can be downloaded and installed from the point-and-click GUI in R, but otherwise most commands and procedures will need to be executed from commands, or code. More advanced users may prefer to use R from their computer's command prompt (Windows) or terminal (Mac OS X and Linux), especially users interested in sequencing and other high throughput

biological data. Individual R commands can be entered one at a time into an R terminal window, but most R users write their commands into a sequential “script” that combines multiple commands in sequence to create an analysis pipeline. Several R scripts are included as supplementary files in these protocols; these R scripts can be opened using the R software itself or using any typical text editor (e.g. Microsoft Notepad, Apple TextEdit, TextWrangler, Emacs, VIM). An introduction to R manual is available to download (<https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>), and I list a few crucial R commands below. Note, sample code will be typeset in Courier New font, also “foo” is a placeholder representing any variable name you might choose and “file.txt” could represent any file name.

- Basic arithmetic
 - Addition and subtraction: `2+4`, `6-3`, `foo+17`
 - Multiplication and division: `2*3`, `8/4`, `foo*7`
 - Powers, logs and exponents: `2^3`, `2**3`, `log10(5)`, `log(6.2)`, `exp(2.1)`
 - Inequalities: `foo == 12.3`, `foo > 5.7`, `foo <= 6.1`
- Search the help menus
 - Learn about a known function (e.g. `t.test`): `help(t.test)` or `?t.test`
 - Search for keyword (e.g. “robust”): `help.search("robust")` or `?? robust`
- Import data from tab-delimited or comma separated value text files
 - `foo = read.delim("file.txt")`
 - `foo = read.csv("file.csv")`
 - `foo = read.table("file.dat", sep = ";")`
- Use specific columns or rows from a data table
 - Use column “Age” from table “dat”: `dat$Age`
 - Use column 3 from table dat: `dat[,3]`
 - Use row 6 from table dat: `dat[6,]`
 - Use row 2 and column 4 from table dat: `dat[2,4]`
- Identify the class (e.g. character, numeric, factor, logical) of a given variable `foo`
 - `class(foo)`
 - `is.numeric(foo)`, `is.character(foo)`, `is.logical(foo)`
- Add comments to annotate or explain R code
 - `2 + 3 # Addition`

- Anything after the “#” symbol is a “comment” that will not be evaluated as code

Wherever possible, an alternate protocol will be shown using GraphPad Prism software (<https://www.graphpad.com/scientific-software/prism/>), which is a commercial statistics software designed for bench scientists and popular in the field of immunology. With some slight modifications, the concepts discussed here would also apply to other statistics software platforms like SAS, JMP, SPSS, Stata, Matlab, Minitab, etc. In the section on statistical design, I will use a few examples from JMP Statistical Discovery software (www.jmp.com).

In addition to choosing your statistics software, you may want to consider acquiring some basic statistics textbooks to serve as references for further study and investigation. Of course, some statistics textbooks and publications will be cited in this Unit wherever appropriate, but anyone reading these protocols may also find some value in keeping 1–2 basic statistics textbooks on hand to answer questions that may not be addressed in these protocols. I’ve included a list of 5 recommended textbooks in the section **Key Resources**.

Basic Protocol 1. Best Practices for Statistical Graphs

Graphs and figures are probably the most elementary way that any reader might interact with statistics and data. Producing a graphic can be deceptively simple using modern software, yet there are many pitfalls that can yield ineffective or even misleading figures. In general, the figures in any scientific publications should follow these 3 ideals:

- Show as much of the data as possible
- Consider the reader and the medium
- Strive for consistency

Showing the data seems like an obvious thing, but we often find reasons to obscure the real data using summaries like barplots or regression lines. Sometimes, it might be appropriate to summarize data, when the data are too numerous or too complicated for a cogent graphic. But in other cases, researchers present overly summarized data either out of habit or convenience or ignorance. Worse, some might purposely obscure data for expediency or even fraud. As an ideal, it is best to show as much of the real data as is possible.

Barplots can be especially problematic figures when they are used to display continuous data, like mean parasite density, mean concentration of a protein or mean percentage of CD4⁺ cells in a sample. Barplots obscure the actual distribution of sample data, even when paired with standard deviations or especially standard errors. Standard errors can be defined in multiple ways, but most standard errors are a ratio of the standard deviation divided by the square root of the sample size (i.e. $se = sd/\sqrt{n}$ for sample size n). Standard errors are always smaller than standard deviations, and many standard errors can be quite small. It’s possible to imagine a case where a barplot with approximately equal standard error whiskers (Fig. 2a) might actually represent samples with wildly different standard deviations (Fig. 2b). More commonly, a barplot with small standard error whiskers might give the illusion of very small variances even when the variances of the raw data are quite large (Fig. 2b).

Another important issue when summarizing data is the proper choice of summarizing metric, usually a mean or median. A barplot representing the means and standard errors (Fig. 2c) might give one picture of the differences between groups (mean A = mean B, mean C = mean D), while a boxplot of the data (Fig. 2d) showing sample medians and quantiles might yield a very different interpretation of the same data (median A = median B, median C = median D). The appropriate metric for a given data set will depend on the distribution of the data and the statistical hypothesis being tested. For example, you would want to display means and standard deviations whenever data are approximately normal and hypotheses are tested using ANOVA or Student's t-test. You would want to use medians or boxplots whenever data are non-normal and nonparametric tests are used.

Beyond merely showing the data and choosing the correct metric to summarize the data, it's also important to consider your reader and the medium in which you are publishing. One classic example is the difference between pie charts (Fig. 2e) and barcharts (Fig. 2f) for summarizing counts from a categorical variable. For years, many statisticians have argued against the use of pie charts in quantitative research, because it is difficult for the human eye to evaluate small differences in the area of a wedge within the pie chart (Fig. 2e). In contrast, it's relatively easy to compare small differences in the height of bars in a bar chart when they represent the counts or percentages of different categorical outcomes (Fig. 2f).

It's also important to consider the medium in which you will be publishing. Will your graphics be printed on paper, or shared electronically and viewed on a computer screen? Can you publish figures in color? How large or small will the figure be? Can the average person read the smallest font? Can your readers distinguish between square and circle symbols or between dashed and dotted line types? Multiple colors or symbols or line types can be a great tool to distinguish different groups in a plot, but are they always necessary? Do they always serve the figure? We can replot the exact same data from Fig. 2d using different colors and symbol types for each of the four groups (Fig. 2g), but does it improve the graphic, or is it a distraction? Dense, complicated figures can be a tool to convey lots of information in small amount of space, but they can also obscure and distract from the very discoveries we want to share (Fig. 2h).

Any scientific publication or presentation is likely to have multiple figures, and it is important to try to present multiple figures consistently throughout any work. Where ever possible, you should try to use the same colors or symbols to represent the same groups throughout your paper. It can be confusing if red symbols mean one thing in the first figure, but something completely different in the fifth figure. Likewise, it's important to use the same type of figure for the same type of data throughout the paper. Very often you'll see a multiple panel figure or consecutive figures showing the exact same type of data, but with different graph types. For example, you might see a page full of multiplex cytokine assays displayed with all scatterplot figures except for one barplot (Fig. 2i,j,k,l). As a statistician and as a reviewer of other people's manuscripts, I know that a manuscript or presentation might represent years of work from multiple contributors. Maybe the one inconsistent graph was created months or years before the other figures. Maybe it was created by a different collaborator on the project. There are many reasonable explanations why one figure might be presented with a different type of graph than other figures representing the exact same

type of data. However, any reasonable reviewer might also be concerned that a different figure type was used to obscure data or mislead readers into thinking a difference is more substantial or less meaningful, depending on what result best fits their narrative. We want to believe the best in people, but inconsistent figures can raise the suspicion of a biased presentation or even fraud.

Examples

Statistical Graphs in R—The download instructions and some very basic examples of critical commands for the R software package were presented in the Strategic Planning section of these protocols. Most of the examples presented in the rest of this manuscript will use R. The R script in supplementary file 1 starts with a list of four library commands, each of which loads a different R package library that adds functions and commands necessary to generate the panels in Fig. 2. In this specific case, the `ggplot2` library is used to generate most of the graphs in Fig. 2, while the `ggpubr` and `gridExtra` libraries are specifically used to create and organize the multiple panels of Fig. 2. The `latticeExtra` library is used to generate Fig. 2h. You can run the entire R script at once using the `source()` command or point and click menu options. Alternatively, you can open the script in your R editor window and run the code one or two lines at a time to better understand how each line of code works.

Each individual panel of Fig. 2 is created using data entered within the R script code, either by directly specifying a vector or matrix of data, or by randomly generating data with R commands. The purpose of each command is usually described with a `#` comment in the code. In the specific case of Fig. 2a,b some of the data are specified in vector format using the `c()` command, while other data are generated randomly using `rnorm()`. The data are later stacked and labeled into a table. Means, standard deviations and standard errors are all calculated from the data to help plot means and error whiskers over the data. The individual graphic panels are created using commands from the `ggplot2` library.

Figures created using `ggplot2` are built one element at a time, and due to the nature of this code it would be difficult or impossible to annotate every command within a specific figure panel. However, generally speaking, the command `ggplot()` creates a new `ggplot` object to be graphed, while the command `aes()` creates a series of aesthetic mappings to be used by the elements of that `ggplot` object. For example, the `aes()` command could define the *x* and *y* variables and the color schemes of a graphic. New elements are added one at a time to the `ggplot` object, including graphic themes (general settings for the appearance of the plot axes, backgrounds, etc.), plot labels, axis limits, and plot elements like bars, error whiskers, mean lines, etc.

One nice property of the `ggplot` graphics is that you can add these elements one-at-a-time and view how each additional element changes the figure. This can help you figure out what each `ggplot` command adds to the graphic. For example, Fig. 2a is constructed using 6 `ggplot` commands: `ggplot()`, `theme_classic()`, `labs()`, `expand_limits()`, `geom_bar()` and `geom_errorbar()`, respectively. You can highlight each command individually and run the code one at a time to see how each command works.

The subsequent blocks of code for Fig. 2c,d or Fig. 2e,f or Fig. 2g are all structured similarly to the code for Fig. 2a,b. Each block of code starts with the commands to generate the data set, then finishes with the code to generate the desired figures. Fig. 2h is unique, because it is a 3D barplot figure that cannot be created using the `ggplot()` library. Remember that R is an open source software created by collaborating researchers from all over the world. The creator of the `ggplot2` package library did not create code for 3D plots, so this panel had to be created using other tools. First, the code creates a completely empty `ggplot()` object to leave an empty space in the multipanel figure, then the `latticeExtra` library is used to create the actual 3D barplot using other graphic devices in R. The resulting 3D barplot was then added into the final multi-panel graph using image manipulation software (e.g. Adobe Illustrator or Photoshop).

Statistical Graphs with GraphPad Prism—Most of the figures in this section can also be created using GraphPad Prism (Supplementary File 2). The first and most important step with Prism is choosing the appropriate Prism data table type for the graphics and statistical tests you want (https://www.graphpad.com/guides/prism/7/user-guide/index.htm?using_prisms_data_table.htm). Choose the column table for t-tests and one-way ANOVA tests with beeswarm plots, or nonparametric tests with Tukey boxplots. Choose the XY table for correlations or regression models with regression line plots or curve fits. Choose contingency tables for Fisher's exact test and barplots, the grouped table for two-way ANOVA with interleaved scatterplots or the survival table for Kaplan-Meier curves.

Once you have data entered into a Prism table, you'll need to choose your graphic type from a list of graphs for your table type (<https://www.graphpad.com/guides/prism/7/user-guide/index.htm?graphs2.htm>). If you are unhappy with the graphic type, you can always change it after the fact. Once you have opened a graphic file, you can edit the figure by clicking in one of two areas on the graph. Click in the middle of the graph or on a specific datum (e.g. symbol, bar, boxplot) to edit the data visualizations in the graph (https://www.graphpad.com/guides/prism/7/user-guide/index.htm?the_format_graph_and_format_ax.htm). Change colors or symbol shapes, change error whisker calculations or data summaries (e.g. mean, median, geometric mean), etc. Click any axis on the plot to change the axis features, including axis tick values, axis labels, figure size, etc. (https://www.graphpad.com/guides/prism/7/user-guide/index.htm?the_format_axis_dialog.htm).

Under the sheet commands, you can create new data tables or duplicate sheets, including new graphic sheets (https://www.graphpad.com/guides/prism/7/user-guide/index.htm?new_graph_of_existing_data.htm). This can allow you to create two different types of figures for the exact same data set, either to compare figures or possibly to overlay two figures on top of one another. The sheet menu also includes the "Page Layout" file type (https://www.graphpad.com/guides/prism/7/user-guide/index.htm?page_layouts.htm), which allows you to arrange multiple graphs on the same sheet. You can either create multi-panel graphics for publication, or you create overlay figures that combine two graphic types (e.g. a beeswarm plot overlaid on top of a boxplot). There are automated commands to equalize graphic sizes, line up figures by their X- or Y-axes, group or ungroup figures together, etc.

Basic Protocol 2. t-tests, ANOVA and Nonparametric Tests

Many immunological experiments will require very simple comparisons of sample measurements between 2 or more groups (e.g. comparing the total amount of IgG between 10 WT and 10 KO mouse samples). In these situations, researchers will often use a simple Student's t-test or a nonparametric Mann-Whitney test, but how should you choose between these two tests? How do you know if a more complicated test is required? What about all the options you see in your statistics software? The following section will answer these questions and more.

The Student's t-test (Fig. 3a,c) is a comparison of the means between two groups (Student 1908, Utts and Heckard 2014). It uses the sample means and the sample errors (a function of the sample standard deviations and sample sizes) to determine whether the population means are likely to be different. Student's t-test assumes its data are normally distributed with independent errors and approximately equal errors (or standard deviations) between the two groups. The t-test calculation starts with a null hypothesis about the population means (e.g. $H_0: \mu_1 = \mu_2$, or the population means are equal) and an alternative hypothesis about the population means (e.g. $H_a: \mu_1 \neq \mu_2$, or the population means are not equal). Next, all the sample data are summarized by the test statistic T^* (equation 1), which is the difference between the two sample means ($\bar{X}_1 - \bar{X}_2$) minus a null value (usually zero) and finally divided by the standard error (i.e. a function of the standard deviations, s_1 and s_2 , and the sample sizes, n_1 and n_2). The test statistic T^* is then compared against a bell-shaped t-distribution to calculate the p-value, which represents the probability of observing these particular data (as summarized by T^*) if the null hypothesis is true. A p-value is usually considered statistically significant if $p \leq 0.05$, and a significant p-value is considered evidence to reject the null hypothesis in favor of the alternative. The t-test is considered a parametric test, because it concerns a parameter (i.e. the mean) of the population distribution (i.e. the Gaussian or "normal" distribution).

$$t^* = \frac{\bar{X}_1 - \bar{X}_2 - 0}{\sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}} \quad (\text{equation 1})$$

Three important concerns with the Student's t-test are (1) paired vs unpaired tests, (2) one-sided vs two-sided tests, and (3) Welch's correction for unequal variances. Whenever the two samples in a t-test represent matched pairs (e.g. longitudinal studies, pre-treatment and post-treatment, left hand vs right hand), it's best to use a paired t-test (Fig. 3c), which is more likely to result in a statistically significant result under matched pair conditions. The paired t-test first calculates the paired differences between all the paired samples, then the paired t-test determines whether the mean of those paired differences is significantly different from zero. Accounting for the paired difference removes the variability due to pairing from the standard error of the t-test, which should result in more significant p-values when data are paired. Generally, it's best to compute two-sided t-tests in all situations, but sometimes we are interested in a more specific alternative hypothesis like $\mu_1 > \mu_2$ or $\mu_2 > \mu_1$.

For example, we might hope that a new drug reduces the average patient's parasite load ($H_a: \mu_{\text{placebo}} > \mu_{\text{drug}}$) and we might not be interested in cases where the drug increases the patient's parasite load ($H_0: \mu_{\text{drug}} \leq \mu_{\text{placebo}}$). In these cases, when you can choose a one-sided alternative hypothesis based solely on your biological expectations BEFORE you collect any data, it is possible to use a one-sided alternative hypothesis for a little extra statistical power. **It is not appropriate to choose the one-sided hypothesis AFTER looking at the data, or worse after looking at your p-values.** Finally, the Welch correction provides a small correction to the standard error calculation for cases where the two sample standard deviations are substantially different. It makes the t-test a little more conservative (i.e. less likely to be significant), but it's an appropriate adjustment when variances are unequal.

Another important consideration is the relationship between sample size, model fit and statistical significance. The Student's t-test is built around a statistical concept called the central limit theorem, which essentially says that the estimate of a population mean gets more and more precise as sample sizes increase and become very large. The central limit theorem has two consequences for the Student's t-test. First, with a large enough sample size, any Student's t-test will become statistically significant, no matter how small the difference in sample means might be. This is the relationship between sample size and statistical power. Because statistical significance has such a strong relationship with sample size, you should always report an effect size (e.g. the size of the difference in means between two groups) with your p-values. Statistical significance tells you whether a result could have occurred by random chance, but the effect size tells you whether a result is biologically meaningful. Second, with very large sample sizes, the assumption of normal data becomes less and less important, sometimes even completely irrelevant. How large is a "large sample size"? The distinction between a large and small sample size is subjective, but historically statisticians have often considered samples $n \geq 30$ to be large for the Student's t-test, since the t-distribution becomes almost indistinguishable from the normal Gaussian distribution at sample size $n = 30$. For other statistical tests, a "large sample size" might be much larger than $n = 30$.

The Mann-Whitney (MW) test, also known as the Wilcoxon Rank Sum test, is a comparison of the sums of the ranked data between two groups (Wilcoxon 1945, Hettmansperger and McKean 1998, Utts and Heckard 2014). Under most conditions, the MW-test acts like a comparison of the sample medians between two groups (Fig. 3b), since in most cases the middle-ranked sample is equal to the sample median. The one exception, where the MW-test does not approximate a comparison of the sample medians, is when the two samples come from very different shaped distributions, particularly one left-skewed sample and one right-skewed sample. In these cases, it is possible to get a significant MW-test p-value even when the sample medians are very similar or even identical. Please note that the MW-test has nothing to do with the sample mean, therefore any graphs associated with the MW-test need to display medians or boxplots (i.e. a representation of the minimum, 25th percentile, 50th percentile or median, 75th percentile and max). Just like t-tests, there are typically options for paired (i.e. the Wilcoxon Sign Rank test) and one-sided MW tests. The MW-test and

paired Wilcoxon Sign Rank test are considered nonparametric tests, because the sum of ranks is not a parameter of any assumed population distribution.

How do you choose between parametric t-tests and nonparametric alternatives like the MW-test? First and foremost, look at graph and visually try to determine if the sample data are approximately normal. Mostly, you are looking to see if the data are unimodal (i.e. the data has one peak) and symmetric. If the data are bimodal or multimodal (i.e. two or more peaks) or if the data are very clearly skewed, then it may be best to choose the nonparametric test. There are statistical goodness-of-fit tests (e.g. Shapiro-Wilk, Kolmogorov-Smirnov, D'Agostino's test) that can determine whether your data are "normally distributed", but these goodness-of-fit tests are all relatively inefficient that require relatively large sample sizes. This means that small sample sizes are unlikely to generate significant results, no matter how non-normal the sample data might be, while large sample sizes are likely to get significant p-values even when the data appears sufficiently normal. That's why I recommend looking at the data plots instead. Note that in some situations it may be appropriate to transform the sample data (usually with a log-transformation) to make the data more normal for a Student's t-test. But when in doubt, it is generally good idea to choose the nonparametric test. Cynically, one advantage is that the nonparametric test will generally receive less scrutiny during peer review, because biological journals often recommend checking the assumptions of t-tests, but not nonparametric tests. The exceptions where you will be better served by the Student's t-test are experiments where it is better or more convenient to compare means than medians, like gene expression experiments, or experiments with variables that are historically considered "normal" (e.g. IQ scores, SAT test scores, human height, etc.). In cases where neither the t-test nor MW-test are appropriate, you can also consider advanced methods like re-sampling tests, which will be discussed later in these protocols.

Often an experiment will require the comparison of three or more groups, which means your statistical analyses will have to account for multiple comparisons. Most researchers are familiar with the idea that a significant p-value is $p < 0.05$, where $\alpha = 0.05$ is often considered the critical level or the Type I error rate. It's always possible that the decision reached after a statistical test might be wrong. If we reject the null hypothesis when it was actually true, that is called a Type I error or false positive (Table 1). If we fail to reject the null hypothesis when it was actually false, that is a Type II error or false negative (Table 1). We usually set up our statistical tests to ensure a low (5%) probability of a Type I error. When we compute multiple statistical tests, each test has its own 5% risk of a Type I error, meaning our probability of making at least one Type I error increases as we compute multiple tests. Think of each statistical test like a lottery ticket, with each additional test giving you a slightly higher probability to "win" a Type I error. When comparing 3 or more groups, we need to adjust our calculations to keep the risk of any Type I errors reasonably low.

Analysis of Variance (ANOVA) is a statistical test that compares the sample means of 3 or more groups in a single statistical test (Utts and Heckard 2014). Just like the t-test, the ANOVA is calculated with a single test statistic F^* (equation 2) that summarized the entire data set with one number. The F-statistic is a ratio of the variability between groups (i.e.

Sums of Squares Difference or SSD) over the variability within groups (i.e. Sums of Squares Error or SSE), each divided by their own degrees of freedom (df) approximately representing sample sizes. When the variability between groups is large relative to the variability within groups, the ANOVA F-test is statistically significant (Fig. 3e). When the variability within groups is larger than the variability between groups, the F-test is not significant (Fig. 3d).

$$F^* = \frac{\sigma_{Between}^2}{\sigma_{Within}^2} = \frac{\text{Mean Square Difference}}{\text{Mean Square Error}} = \frac{\text{Sums of Squares Difference}/df_1}{\text{Sums of Square Error}/df_2} \quad (\text{equation 2})$$

The initial ANOVA F-test only tells you whether some of the groups are different from one another, but it doesn't specify which groups are different from one another, therefore a significant ANOVA F-test is usually followed up with post hoc tests to compare specific groups. The post hoc tests are usually adjusted to account for multiple comparisons. There are several types of adjustments available, but the most popular adjustments are Tukey's Honest Significant Difference (HSD) for all pairwise comparisons, Dunnett's Multiple Comparisons against Control (MCC) for comparisons of all groups against a single control and the Bonferroni adjustment for custom comparison lists. The Tukey and Dunnett adjustments are more powerful than Bonferroni for their respective comparisons, but the Bonferroni adjustment is easy to compute and very flexible when custom lists of comparisons are needed.

The ANOVA model is very flexible, allowing for many types of complicated tests. There are repeated measures ANOVA models, where paired observations can exist among 3 or more time points or conditions (Fig. 3g). There are multifactor ANOVA models, where multiple types of groups can be compared with a single test, for example a two-way ANOVA model comparing %Atypical B-cells in patients identified as PCR+ and PCR- for *Pf* infection at three time-points (pre-challenge, 3 days post and 7 days post). Any two factors in an ANOVA can be independent, or crossed, effects, where all possible combinations of the two factors exist. For example, gender and infection are crossed effects, with all four combinations (1) female uninfected, (2) female infected, (3) male uninfected and (4) male infected can exist. Alternatively, two factors can be dependent, or nested, where the groups, or levels, in one factor depend upon the other factor. For example, you might compare 3 doctors each from 2 different hospitals: Dr. Smith, Dr. Wu and Dr. Gonzales from Liberty Hospital and Dr. Jones, Dr. Traore and Dr. Li from Advanced Medical Center. There are mixed model ANOVAs that distinguish between fixed effects (e.g. PCR+ vs PCR-, Male vs Female) and random effects (e.g. differences from patient to patient, differences from hospital to hospital, differences among multiple incubators, etc.). The ANOVA model is highly adaptable, and it would be impossible to cover every variation on the ANOVA model in this protocol review, but hopefully this has provided a useful primer on the model and its most popular variations.

Just like the Student's t-test, there are also nonparametric versions of the ANOVA model. The Kruskal-Wallis (KW) test is a nonparametric extension of the Mann-Whitney test for

single-factor comparisons of 3 or more groups (Utts and Heckard 2014). Just like ANOVA, the KW-test must be followed up with properly adjusted post hoc Dunn's tests to make pairwise comparisons among the individual groups. Friedman's test is another variation on the MW and KW tests for single factor comparisons with repeated measures pairing among groups. Unfortunately, there are no simple methods to extend the nonparametric tests into multifactor experiments.

Examples

t-tests, ANOVA and Nonparametric Tests in R—The Student's t-test and Mann-Whitney test can be computed in R using the commands `t.test()` or `wilcox.test()`, respectively, at the command line (Supplementary File 3). Alternatively, you can save the results to a variable of your choosing (e.g. `foo = t.test()` or `bar = wilcox.test()`), so that you can recall specific test results like the p-value in subsequent calculations (Supplementary File 3). The tests can be computed by specifying two different columns of data (e.g. `x = PCRneg`, `y = PCRpos`) or by specifying a formula and a data set (e.g. `formula = TotalIgG ~ Infection`, `data = dat2a`). Remember to match t-tests with mean plots (Fig. 3a), Mann-Whitney tests with medians or boxplots (Fig. 3b). Paired t-tests and Wilcoxon tests can both be matched with before and after connected line plots (Fig. 3c).

Computing ANOVA models in R typically requires at least two lines of code, often more. At the very least the basic ANOVA model needs to be fit using `lm()` command, then the results can be viewed using the `anova()` or `summary()` commands. Pairwise post hoc contrasts like Tukey's HSD can be computed using the `glht()` command from the `multcomp` package library. Kruskal-Wallis tests are computed using the `kruskal.test()` command, with Dunn's post hoc tests computed using `dunn.test()` from the `dunn.test` library. Repeated measures ANOVA can be computed in many ways, but one of the most common is with a mixed model using the `lmer()` from the `lme4` library. Friedman's test for nonparametric repeated measures uses the `freidman.test()` command, with post hoc tests from the `PMCMR` library.

t-tests, ANOVA and Nonparametric Tests in GraphPad Prism—In GraphPad Prism (Supplementary File 4), first enter your data in a column data table, then compute Student's t-tests, MW-tests and paired tests by clicking > **Analyze > Column analyses > t tests (and nonparametric tests)** (https://www.graphpad.com/guides/prism/7/statistics/index.htm?analyses_of_one_grouping_varia.htm). Options for Unpaired and paired experimental designs, and parametric or nonparametric tests will be immediately available on the Experimental Design tab. If the unpaired parametric test (Student's t-test) is selected, you will have the option to select Welch's correction for unequal standard deviations. A second tab of Options allows you to select between one-tailed and two-tailed tests, select different confidence levels (e.g. 90%, 95%, 99% confidence), or create additional plots and output.

Single factor ANOVA (with or without repeated measures), Kruskal-Wallis and Friedman's test can be computed from column tables in GraphPad Prism by clicking > **Analyze > Column analyses > One-way ANOVA (and nonparametric)** (<https://www.graphpad.com/>

[guides/prism/7/statistics/index.htm?stat_comparing_three_or_more_groups.htm](https://www.graphpad.com/guides/prism/7/statistics/index.htm?stat_comparing_three_or_more_groups.htm)). Just like the t-test analysis, the Experimental Design tab allows you to choose between matched pairs repeated measures or no matching, parametric (ANOVA) and nonparametric tests. It is also important to check the Multiple Comparisons tab to select your post hoc comparisons (https://www.graphpad.com/guides/prism/7/statistics/index.htm?stat_multiple_comparisons_after_ano.htm). Tukey (each column with every other column), Dunnett (each column vs control column), Bonferroni tests (preselected pairs) and Dunn's tests (nonparametric) are all available on the Options tab, along with a few alternative procedures you can read about in the Prism help menus. Prism can also compute two-factor ANOVA with or without repeated measures from the slightly more complicated grouped table type (https://www.graphpad.com/guides/prism/7/statistics/index.htm?two-way_anova.htm). The group tables organize one factor on the columns, and the second factor on the rows, with replicate values entered in subcolumns within each column factor level. Click > **Analyze > Grouped analyses > Two-way ANOVA** to fit the two-way model with several options for matching on the Experimental Design tab and familiar options post hoc multiple comparisons on the Multiple Comparisons and Options tabs.

Basic Protocol 3. Linear and Nonlinear Regression

Linear Regression—Often in an immunological experiment, researchers will need to quantify the relationship between two (or more) quantitative variables. For example, you might want to estimate a linear correlation between a patient's age and the total IgG against a pathogen. You may also want to model the relationship between the dilutions of plasma serum and the optical density (OD) response in an Enzyme-Linked ImmunoSorbent Assay (ELISA). Sometimes a simple correlation coefficient is sufficient to describe a relationship, while other relationships might require a more complicated statistical model and close attention to the model fit. The following section will describe some basic concepts behind correlations, linear regression models and nonlinear regression models.

Correlation coefficients provide a simple measure of the association between two quantitative variables (Fig. 4a–d). Almost all correlation coefficients range between -1 (perfect negative correlation) and $+1$ (perfect correlation), with 0 representing no correlation at all. The Pearson correlation coefficient (r) is the most commonly used; anytime you see an unlabeled correlation value (or correlation squared, r^2) in a publication, you can generally presume the calculation is Pearson's. The Pearson's correlation assumes the two variables have a linear relationship and the data come from a bivariate normal (i.e. Gaussian) distribution, meaning that not only are the data from the two individual variables each normally distributed, but the combination of the two variables together is also normally distributed. You can see 3 examples of bivariate normal data with zero correlation (Fig. 4a), weak positive correlation (Fig. 4b) and strong negative correlation (Fig. 4c) for reference.

Spearman and Kendall rank correlations are used whenever data are not bivariate normal, or more often when correlations are not linear (Fig. 4d). Briefly, the Spearman rank correlation coefficient is simply the Pearson correlation coefficient computed on the ranks of the data, while the Kendall rank correlation coefficient is a measure of the relative concordance or

discordance of the ranked data. For example, if the ranked data for the variable x is sorted 1, 2, 3, 4, 5, then the matched ranks of variable y could yield:

- Perfect concordance: 1, 2, 3, 4, 5 (Kendall tau = +1)
- Strong concordance: 1, 2, 3, 5, 4 (Kendall tau = +0.8)
- No concordance: 4, 1, 3, 5, 2 (Kendall tau = 0)
- Strong discordance: 5, 4, 3, 1, 2 (Kendall tau = -0.8)
- Perfect discordance: 5, 4, 3, 2, 1 (Kendall tau = -1)

Beyond simple measures of correlation, often statisticians or researchers will want to compute a regression to model the specific relationship between two variables (Fig. 4e). Regression models can be used to make predictions and they offer additional diagnostics to assess the fit of the model, unlike simple correlations which only offer an estimate of the magnitude and direction of the relationship and maybe a p-value to estimate its significance (Utts and Heckard 2014). Most regressions you encounter in biological literature will be some form of linear regression. The term “linear regression” can be a little confusing, because it doesn’t always refer to straight line relationships. Instead, a linear regression refers to any regression model that can be expressed using the familiar algebraic equation of a line $y = mx + b$, where x represents an independent variable, y represents a dependent variable, m represents a model slope and b represents the y-intercept (Kutner *et al.* 2004). In statistics, we usually express the simple regression model using equation 3, and more complicated regression models using matrix algebra in equation 4.

$$y = \beta_0 + \beta_1 x + \varepsilon \quad (\text{equation 3})$$

$$y = x\mathbf{b} = [1 \ x_1 \ x_2 \ x_3 \dots x_k] \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_k \end{bmatrix} \quad (\text{equation 4})$$

Just like the ANOVA model, linear regression is a very flexible model with far too many options and concepts to cover in one relatively short article. One concept critical to linear regression, and ANOVA as well, is the idea of model fit. Model fit is usually assessed by examining the model residuals (Fig. 4f). Residuals, or model errors, are the differences between your observed data and the predictions from the regression model (i.e. the regression line). We usually define the residuals e using equation 5:

$$e_i = y_i - \hat{y}_i \quad (\text{equation 5})$$

where i represents a particular data point (e.g. patient #581), y_i represents the observed data for that data point, and $\hat{y}_i$ represents its predicted value from the regression line. Just like the Student's t-test and ANOVA model, we expect the data from a linear regression to be approximately normally distributed with constant variance through the range of the independent variable x , which implies that the residuals should also be normally distributed with constant variance, centered around zero. Typically, statisticians assess model fit using a plot of the residuals vs fitted values (Fig. 4f) or residuals vs X values (Supplementary File 6). The model fits well if the residuals look like a random shotgun blast, or maybe a rectangular or American football shape. If the residuals form a triangle or cone shape, that indicates the model has non-constant variance (i.e. variance increases or decrease as X increases), which is a violation of the model assumptions. Likewise, if the residuals form a curved shape or a regular pattern, it might be an indication of poor model fit, possibly a missing predictor variable from your regression model.

There are many other indicators of model fit, besides the residuals, but maybe the next most important is the model leverage. We hope all the data points have an equal impact on the fit of the regression model, but data points at the far minimum or far maximum of the X-values can have a disproportionate influence on the regression. This is called leverage, because these extreme X-values can pull the regression line away from the rest of the data, like a hand at the end of a long lever (Fig. 4g). We can calculate an estimate of the amount of leverage exerted by each individual data point and plot them to identify data points that might have high influence, which are potential outliers (Fig. 4h). If you are interested in other measures of model fit, you may want to check a statistics textbook to learn about Cook's Distance, df fits, df betas, etc (Kutner *et al.* 2004).

Just like the t-test and ANOVA models, it is also possible to fit nonparametric versions of linear regression, notably quantile regression (Koenker 2005, Koenker and Bassett 1978). Where the linear regression fits a line through mean response at each value of X, a quantile regression fits a prediction line through the median or a known quantile of the data at each value of X. The example (Fig. 4i) shows a quantile regression through the 10th percentile of Total IgG, which might be a useful model if your Total IgG response variable was not normally distributed at each value of X. In the example, we see a strongly right-skewed response, with the majority of the data falling at the bottom of the regression line and long tails reaching upward, making the quantile regression model a reasonable choice. More typically, you might fit a quantile regression to the 50th percentile, aka the median. Transformations of the Y-variable are another popular option when data are not normally distributed.

The regular linear regression model can be extended in many ways, but the most common way is to compute a regression model with multiple independent variables, X , and a single response variable, Y , in a multiple regression. When multiple continuous X-variables are used, the resulting model is also called a regression surface. If two X variables are used to predict a single Y variable, the model can be visualized as a 3-dimensional surface. Supplementary File 5 includes code to generate such a regression surface, then plot the data in an interactive 3D object that can be rotated or flipped in any direction (not shown, must

run the code in R). Regression models with more than 2 X predictors are still called regression surfaces, but obviously cannot be visualized as easily.

It is also possible to compute a regression model with one (or more) continuous X variable(s) and at least one categorical X variable (Fig. 4j). The resulting model is typically called an Analysis of Covariance (ANCOVA), or sometimes a regression with dummy variables. These models produce multiple regression lines or multiple regression surfaces that can be compared against one another. When described as ANCOVA, these models often place a higher emphasis on comparing data between categorical groups (e.g. drug treatment, infection status), like an ANOVA model with the continuous X-variables included only as an adjustment for some nuisance variable(s) that may be viewed as less important (e.g. patient age or weight).

One last extension of the linear regression is polynomial regression, which is a nice segue to the topic of nonlinear regression. The polynomial regression model (Fig. 4k) adds one or more exponential powers of a predictor variable x to add one or more curved bends to the regression line. Each added power of x adds another curve to the polynomial line.

- Quadratic regression (single curve): $y = \beta_0 + \beta_1x + \beta_2x^2 + \epsilon$
- Cubic regression (two curves): $y = \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3 + \epsilon$
- Quartic regression (three curves): $y = \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3 + \beta_4x^4 + \epsilon$
- ...

Polynomial regression is still considered a linear regression model, because the equation still matches the general format of a linear equation, but it allows researchers to fit a curved trend to their data. If there are k unique values of the predictor variable x , then the maximum polynomial model has $k - 1$ terms (e.g. $X = \text{age}$ with six unique values 2, 3, 4, 5, 6 and 10 years old, the maximal polynomial is 5th order). Interestingly, if you fit the maximum number of polynomial power terms allowed by your data, the polynomial regression model becomes similar to the ANOVA model, with each unique X-value having its own curve and acting almost as a factor level group. Polynomial regression is a powerful method to help fit a curve to data, even when you do not have a defined equation to model the biological process.

Non-linear regression—The final regression topic to introduce is the idea of nonlinear regression. Excluding quantile regression, all the previously discussed linear regression models could be fit using ordinary least squares (OLS). Essentially, the simple nature of the linear equation ($y = mx + b$) used in these regression model allowed statisticians to identify the best estimate of the regression slopes and intercepts using a single step of calculus. True nonlinear regressions use a modeling equation that is not linear (i.e. not of the form $y = mx + b$). For example, immunologists will often use the logistic dose-response curve (Fig. 4l) in equation 6 to model biological assays like Enzyme-Linked Immuno-Sorbant Assay (ELISA).

$$Y = Bottom + \frac{Top - Bottom}{(1 + 10^{Slope(LogEC50 - X)})^{Symmetry}} \quad (\text{equation 6})$$

In this nonlinear model, the terms Bottom, Top, LogEC50, Slope and Symmetry are the unknown parameters, similar to the parameters $\beta_0, \beta_1, \dots, \beta_k$ in a linear regression. Typically, it is not possible to identify the best estimates of the parameters in a nonlinear regression using OLS equations. Instead, the parameters of a nonlinear regression are most often estimated using computational methods, often the Levenberg-Marquardt method. These algorithms start with an initial guess about the possible values of the unknown parameters that might fit the curve to your data, then the computer will make a small change to those initial guesses to see if the model fit improves. If the model fit improves, the computer will make another small change in that same direction to see if the fit improves any further. If the model fit decreases, the computer will try a small change in a different direction. Eventually, through a series of small changes to the unknown parameter values, the fit of the model will converge to a solution where any subsequent changes to the parameters will only yield very small changes in the fit of the model and the procedure will stop.

Nonlinear regressions have unique challenges that are a little different than linear regression, due to the computational methods used to estimate the parameters. Often times a nonlinear regression will “fail to converge”, yielding no estimates of the unknown parameters and no p-values to test any hypotheses, either comparing two nonlinear regression lines or comparing unknown parameters (e.g. LogEC50) against possible null values (e.g. LogEC50 = -8). When nonlinear regression models fail to converge, one possible correction is to change the initial values of the equation. If the initial values of the equation are too far away from the final solution, the Levenberg-Marquardt algorithm may not be able to find a solution. Another helpful option is to apply constraints to specific parameters (e.g. Bottom must be greater than 0, Top must be less than 2.5, or Slope must be positive). Often, assays like ELISA require a fixed series of dilutions that do not work well for all samples, yielding some data sets that do not have any data in specific regions of the curve. For example, if you have no data near the top of the curve, the computer will have no idea whether the maximum optical density (OD) might be 2.0 or 8000 or Infinity. Enforcing a parameter constraint can eliminate these problems. The instruction manual for GraphPad Prism contains some especially helpful tips for improving the fit of nonlinear regression models (https://www.graphpad.com/guides/prism/7/curve-fitting/index.htm?reg_nonlinchoices.htm).

Examples

Linear and Nonlinear Regression in R—Simple Pearson correlation coefficients can be computed using the `cor()` command in R (Supplementary File 5), which returns a variance-covariance matrix of the two variables. You will always see correlations equal to 1 on the main diagonal of (1,1), (2,2), etc., because they represent the correlation of each variable with itself (e.g. correlation of variable 1 with variable 1, correlation of variable 2 with variable 2). The actual correlation(s) of interest will be off the main diagonal of the variance-covariance matrix in mirrored entries like (1,2) and (2,1), which of course represent

the correlation of variable 1 with variable 2. You can compute a statistical test to evaluate the significance (i.e. p-value) of a correlation using the function `cor.test()`. Totally uncorrelated (Fig. 4a), weak positive correlated (Fig. 4b) and strong negatively correlated bivariate normal data (Fig. 4c) are shown for reference. Spearman and Kendall rank correlations for nonlinear trends and non-bivariate normal data can be also be computed (Fig. 4d).

Like the ANOVA models in the previous section, linear regression models (Fig. 4e) usually require two or more steps to compute. Most linear regression models can be fit and stored using the `lm()` command, then different results can be viewed using the `summary()` or `anova()` functions. Usually, a researcher is most interested in the estimates and statistical tests of the slope parameters (e.g. Age = 1.41 in Fig. 4e). Note, the y-intercept and slope parameters are stored in the basic model fit, while the ANOVA table (outAOV) and the model summary (outSUM) produce the statistical tests. Alternatively, a researcher might want to predict a response and calculate a 95% confidence interval for a specific value of X using the `predict()` function (e.g. what is the expected Total IgG at age = 3? At age = 7? Age = 12?). Diagnostic measures can also be extracted, computed or saved from the saved “fit” object. Residuals (Fig. 4f), fitted values and other output can be recalled from a saved fit using the `$` operator (e.g. `fit$residuals`). Leverage values (Fig. 4h) can be computed with the function `hatvalues()`, while other diagnostics can be called with their own functions, including `cooks.distance()`, `dffits()` and `dfbetas()`. One regular linear regression line can be plotted using the `geom_smooth()` command from `ggplot`.

Nonparametric quantile regression (Fig. 4i) can be fit with the `rq()` function from the `quantreg` library. The `tau` parameter determines the specific quantile that will be regressed, e.g. `tau = 0.1` for 10th percentile, `tau = 0.5` for 50th percentile (median regression) or `tau = 0.85` for 85th percentile. Note, the quantile regression line needs to be plotted manually using the `geom_segment()` command from `ggplot`, starting the line at the y-intercept `x=0, y=fit$coef[1]` and ending with the predicted value at the maximum value of `x = 16` in this example, with `x=16, y=fit$coef[1]+16*coef$fit[2]`.

A multiple regression can be fit with the `lm()` command by specifying multiple X variables in the `formula` parameter of `lm()` or other regression functions (e.g. `rq()`, `lme4()`, ...). Many formula specifications will work. The formula `TotalIgG ~ Age + Weight` would model a flat surface, while adding interactions (e.g. `TotalIgG ~ Age + Weight + Age:Weight`) or polynomial terms (e.g. `TotalIgG ~ Age + AgeSq`) would add bends and curves to the surface. Note, the formula `TotalIgG ~ Age*Weight` is equivalent to `TotalIgG ~ Age + Weight + Age:Weight`. An interactive 3D plot (not shown in figure, must run R code) of the data points can be made using `plot3d()`, while a 3D surface can be added using `surface3d()`.

Analysis of Covariance (ANCOVA) can also be fit using the `lm()` command by adding both continuous and categorical X variables to the formula (e.g. `TotalIgG ~ PCR*Age`, where Age is continuous and PCR is categorical). The ANCOVA model can be viewed in two different ways, either as a series of independent regression lines to be compared (Fig. 4j), or

as an ANOVA model comparing two or more groups (e.g. PCR– and PCR+) after controlling for a continuous nuisance variance (e.g. Age). Looking at the regression coefficients, you may realize that one of your categorical variable groups (e.g. PCRneg in the example) is treated as the “baseline” group, with the general Y-intercept and slope representing that baseline group. The other parameters (e.g. PCRpos, PCRpos:Age) represent adjustments to those baseline Y-intercept and slope coefficients, respectively.

Statistical tests of any ANOVA interaction terms (e.g. PCRpos:Age) represent comparisons of the line slopes. If the interaction term is significant, then the slope is significantly different from the baseline group. Any comparisons of the predictions from the two lines will depend on the value of the regression variable(s) X. For example, predictions from the lines may be significantly different at Age = 10, but not at Age = 1. Comparisons of the mean response or the predicted response for one new individual can be visualized using confidence intervals (Fig. 4j) or prediction intervals (not shown). If the interaction is not significant, then the line is approximately parallel with the baseline group and you can look at the significance of the Y-intercept terms (e.g. PCRpos parameter). If Y-intercept term is significant, then the line is either significantly higher or significantly lower than the baseline group.

Polynomial regression (Fig. 4k) can also be fit using `lm()`, but it is important to note that each polynomial term should be calculated and stored in the model's data frame (e.g. `AgeSq = Age^2`) then called specifically in the model formula (e.g. `formula = TotalIgG ~ Age + AgeSq`). Simply adding the term `Age^2` to the model will not work in the `lm()` command or most similar R functions. Some statisticians recommend centering the polynomial terms (e.g. `AgeSq = (Age - mean(Age))^2` for better results. Interpreting the terms of a polynomial regression can be difficult, but researchers are often more interested in the fit of the line to the data than the statistical significance of the individual parameters.

Nonlinear regression (Fig. 4l) can be fit using the `drm()` function from the `drm` library. The `fact` parameter of the `drm()` function determines the model fit to the data, in the example the option `L.5()` represents a 5-parameter logistic dose-response curve. Other nonlinear regression model functions include log-logistic (`LL.5`), Weibull (`W1.4`), Brain-Cousens (`BC.5`), Asymptotic Regression (`AR.2`), and Michaelis-Menten (`MM.3`). Please note that different models may expect log-transformed X variables. An example with only one nonlinear regression line is shown, but multiple nonlinear regression lines can be fit and compared.

Linear and Nonlinear Regression in GraphPad Prism—Linear and nonlinear regression methods can be computed and graphed in GraphPad Prism (Supplementary File 6) from the XY table type (https://www.graphpad.com/guides/prism/7/curve-fitting/index.htm?reg__curve_fitting_with_prism_6.htm). Click **Analyze > XY analyses >** to choose from correlation calculations, linear regression models or nonlinear regression models. Opening **Analyze > XY analyses > Correlation** will bring up a single menu where you can choose between parametric Pearson correlations of nonparametric Spearman

correlations. Kendall correlations are unavailable. Please note that correlations can be computed for other table types like column tables.

Clicking **Analyze > XY analyses > Linear regression** will open a single menu where you can choose several regression options (https://www.graphpad.com/guides/prism/7/curve-fitting/index.htm?reg_linear_regression_with_prism.htm). Check the box to “interpolate unknowns from standard curve” and you can make predictions for any unpaired X- or Y-data entered in the XY table. Check the box to “test whether slopes and intercepts are significantly different” to fit an ANCOVA model when multiple Y groups are entered in the XY table. There are options to display confidence or prediction bands at 90%, 95% or 99% levels. You can add a model constraint to force the regression line to go through Y-intercept = 0 or any fixed intercept value.

Click **Analyze > XY analyses > Nonlinear regression (curve fit)** to fit true nonlinear regression models as well as some complicated linear regression models, like polynomial regression (https://www.graphpad.com/guides/prism/7/curve-fitting/index.htm?reg_nonlinchoices.htm). The nonlinear regression menu currently has 10 menu tabs with different sets of analysis options. The model fit tab allows you to choose a regression model type from a list of commonly used nonlinear and linear models. If you are interested in a particular model, highlight the model and click the gear symbol or the question mark button labeled “learn about this equation” to see more details, including the model equation and parameter definitions. Many models have specific biological contexts, like dose-response relationships, ligand binding, enzyme kinetics, etc. There are also options to use regular or robust fit and options to interpolate unmatched data points.

The compare tab allows you to choose one type of statistical test for the model. Generally, you either want to compare the fit of two different regression models to determine which model fits best, or you want to compare the regression curves between multiple Y-value groups entered in the data table. Statistical tests can be computed using ANOVA-style extra sums of squares F-tests, or using Akaike’s Information Criterion (AIC). The F-test method works best when two models are closely related (e.g. cubic polynomial regression vs quadratic polynomial regression; 5-parameter logistic curve vs 4-parameter logistic curve). The AIC tests work better when two models are wildly different (e.g. polynomial regression vs logistic curve).

The constrain tab is used to introduce model constraints, which can help improve model fit when data sets are incomplete or when X-values are not optimized for a specific sample. For example, you may only have data in the bottom half of a logistic dose-response curve because the X-dilutions were too weak for your sample; a model constraint can help fit the top of the curve when there is no data to estimate the top of the curve. Likewise, model weights can help put more emphasis on areas of the curve where you have more information, and less emphasis on areas of the curve that are low information. Supplying better initial values can help difficult models converge, if the model will not fit properly. Other nonlinear regression model options control model output and diagnostic plots.

Basic Protocol 4. Categorical Response Models: Contingency Tables, Generalized Linear Models and Survival

Simple t-tests, ANOVAs and regressions will be appropriate for many, if not most, immunological experiments, but all these methods assume you will have at least one quantifiable, continuous response. What do you do if you want to predict a categorical response? For example, what if you want to evaluate the effects of early season malaria exposure (PCR– or PCR+) on susceptibility to future malaria infection (susceptible or immune)? Or you might want to assess the effects of patient age on their susceptibility to malaria? Survival and reliability analyses are very common analyses for immunological experiments, for example time to first infection or time until death between different vaccine groups or different demographic groups. These kinds of analyses will be covered in this section.

The simplest categorical response models are contingency tables (Utts and Heckard 2014), which can be tested using Pearson's chi-squared test of independence, Fisher's exact test (Fig. 5a,b) and related methods like McNemar's test (paired data, Fig. 5c) or the Cochran-Mantel-Haenszel test (difference between two contingency tables, Fig. 5d). Table 2 shows an example of contingency table with 2 rows and 2 columns (i.e. a 2×2 contingency table). The Pearson's chi-squared test and Fisher's exact test both test whether there is a correlation between the row and column variables of the table. Both tests yield similar results, but the Pearson calculation is much more sensitive to table cell counts that are small (< 5) or zero, therefore I recommend using Fisher's exact test wherever possible. One small advantage of the Pearson's test is that it is easier to extend to larger M×N tables, like the 3 row by 4 column example in Table 3. Fortunately, the `fisher.test()` function in the R Project software can compute Fisher's exact test for M×N tables, but few other statistics software packages can compute Fisher's test for anything but 2×2 tables. For example, GraphPad Prism will only compute Fisher's exact test for 2×2 tables. Table 4 shows an example of paired table data for McNemar's test, while Table 5 shows a three factor table for the Cochran-Mantel-Haenszel test.

Fisher's exact test, Pearson's chi-square test, McNemar's test and the Cochran-Mantel-Haenszel test all produce p-values to evaluate the significance of relationship between the rows and columns of the contingency tables, but usually researchers will also want an odds ratio (equation 7), or sometimes a relative risk (equation 8) calculation to help interpret the table and the statistical result (Utts and Heckard 2014). Please note that the odds ratio (OR) and relative risk (RR) can be computed in at least two different directions; usually you try to choose the calculation that results in $OR > 1$ or $RR > 1$. An odds ratio is usually interpreted by saying something like, "the odds of being immune are 6.62 times higher for PCRneg patients than PCRpos patients". Likewise, relative risk is usually interpreted as, "the relative risk of being immune is 2.82 times higher for PCRneg patients than PCRpos patients". Multiple odds ratios and relative risks can be computed for any pair of response groups and any pair of predictor groups whenever tables are larger than 2×2. Most statisticians prefer using odds ratios, in part, because the odds ratio can be derived from more complicated

generalized linear model parameters. Some medical doctors and biologists may prefer relative risk calculations, because risks are sometimes easier to understand than odds.

$$OR = \frac{Row\ 1A / Row\ 1B}{Row\ 2A / Row\ 2B} = \frac{23/11}{6/19} = 6.62 \quad (\text{Equation 7})$$

$$RR = \frac{Row\ 1A / Sum\ Row\ 1}{Row\ 2A / Sum\ Row\ 2} = \frac{23/(11 + 23)}{6/(19 + 6)} = 2.82 \quad (\text{Equation 8})$$

Generalized linear models (GLM) are more advanced categorical response models that can be applied to a wide variety of problems, including larger contingency tables (Agresti 2012). Technically, the generalized linear model is an extension of the regular linear model used to calculate ANOVA and linear regression, except generalized linear models add the option for new distribution families and link functions. For example, a regular linear regression model could be understood as a generalized linear model with a Gaussian distribution and an identity (i.e. $f(x) = x$) link function. More relevant to categorical data analysis, you can also use a GLM with a Poisson family distribution and a log link function to analyze a contingency table with more than two dimensions, like the $3 \times 2 \times 2 \times 2$ table example in Table 6 (Fig. 5e). Historically, these contingency table GLMs used to be called log-linear models, and they can be confusing because the biological outcome (e.g. Immune or Susceptible status) is treated as another one of the predictor variables, while the model response variable is the “count” in each cell of the contingency table. But while this approach may not be intuitive at first, the log-linear model (i.e. Poisson family GLM) can be used to evaluate complex contingency table relationships with both statistical tests and model odds ratios. Note that these high dimensional tables for complex GLMs are often presented in “tall” format with a single column of counts (Table 6), rather than a “wide” format where counts are distributed among both columns and rows (Tables 2, 3, 4 and 5).

There are many closely related types of generalized linear models to analyze complex contingency table data, including Negative Binomial family models (which are better at modeling variance than Poisson models), zero-inflated models for tables with sparse counts (i.e. lots of zero counts in the table cells) and generalized linear mixed models (GLMM) for tables with random effects (e.g. effects that represent a random sampling, like a sample of 6 different hospitals and clinics). Often multiple generalized linear models are fit to a particular table, then compared using statistical tests based on the deviance residuals or -2 log-likelihood. Alternatively, the model with the smallest AIC might be selected as the best model. The topics of model selection and generalized linear model fit can be quite complicated and you should really consult a statistician if you believe these models might be appropriate for your data, but you feel uncomfortable selecting a specific model.

A different type of generalized linear model that is also commonly used in biology is logistic regression, which is used to predict a categorical response using a continuous predictor

variable. For example, predicting whether a patient will be immune or susceptible to malaria based solely on patient age (Fig. 5f). Just like linear regression, ANOVA and other types of generalized linear model, the logistic model can be extended to account for multiple covariates or factor effects (Fig. 5g), for example comparing two logistic regressions for PCR– and PCR+ subjects.

Survival and reliability models (Agresti 2012) are very commonly used in immunology experiments, since researchers will often want to study the effects of various infections and treatments on model organisms like mice, rats, etc. Note that “survival” methods aren’t solely used to analyze life or death experiments; they can also be used to test other endpoints like “time to first infection”. The most common survival method is the Kaplan-Meier curve (Fig 5h), which describes the time until death or time until event for patients or animals in a study, often with some data that have been “censored” due to subjects leaving or entering the study at irregular times, leaving only partial information about their time to event. Two or more Kaplan-Meier curves can be compared using log-rank tests or by examining the 95% confidence intervals around the Kaplan-Meier curves (Fig 5h).

More complicated survival and reliability models exist, and one of the most popular is Cox regression, or proportional hazards regression, which examines the effect of continuous predictor variables like age or parasite density on subject survival or time to event. The Cox model is sometimes called proportional hazards regression, because any continuous covariate is assumed to have a multiplicative effect on the hazard rate (i.e. risk of dying, risk of event), meaning the effect of a 1 unit increase age is the same from age 2 to 3 or from age 24 to 25. These Cox regression parameters can also be tested. If researchers do not want to assume proportional hazards, there are also parametric survival regression methods that use explicit distributions, like the Weibull distribution. Unfortunately, there are no standard ways to graph the results from Cox regression. Some researchers will simply graph the confidence intervals of each cox regression coefficient. If data sets are large enough, some researchers will break the data set into 3 or more covariate groups (e.g. ages 2–5, ages 6–9, ages 10–15) and plot Kaplan-Meier curves for each covariate group. In my example (Fig. 5i), I have used a color scale of lighter to darker color to represent increasing subject age.

Examples

Categorical Response Models in R—Pearson’s chi-squared test of independence and Fisher’s exact test can be computed in R (Supplementary File 7) using the commands `pearson.test()` and `fisher.test()`, respectively. Just like the `t.test()` and `wilcox.test()` functions presented in the last section, the `pearson.test()` and `fisher.test()` commands can either be called directly at the command line, or their results can be stored in a variable of your choosing, possibly to recall specific results like the p-value. It’s best to compute odds ratios or relative risk estimates by hand. The Fisher’s exact test command `fisher.test()` will compute an odds ratio for 2×2 tables (Fig. 5a), but it is a conditional odds ratio rather than the unconditional odds ratio you would compute by hand. Multiple odds ratios or relative risks are computed by hand for the 3×4 table (Fig. 5b), as shown in the examples.

McNemar's test for paired table data and the Cochran-Mantel-Haenzel test for comparing two 2×2 contingency tables are computed using `mcnemar.test()` and `mantelhaen.test()`, respectively. The McNemar's test command works just like `pearson.test()` or `fisher.test()`, but it assumes the contingency table data are paired in nature. The `mantelhaen.test()` command can be used with either a 3-dimensional array of data (as shown in the example), or with 3 separate factor variables `x`, `y` and `z`.

Just like ANOVA and regression problems, the Poisson family generalized linear model must be fit and evaluated in at least two separate steps. The model is fit to the data using the `glm()` command with parameter `family = poisson`, then the results are viewed using `summary()` and other commands. The model coefficients can be extracted using the `coef()`, then odds ratios can be computed using the exponent function `exp()`. If multiple GLMs are fit to the same data, you could compare model fits using AIC or $-2 \log$ -likelihood. Here, we see one example fitting a negative binomial family generalized linear model using the `glm.nb()` function from the MASS package library, but you could also explore zero inflated models using `zeroinfl()` from the pscl library or generalized linear mixed models using `glmer()` from the lme4 library.

Logistic regression can also be fit using `glm()`, but with `family = binomial` instead of `poisson`. The `summary()` command works the same as for other models. The same tricks to extract model coefficients and compute odds ratios from the Poisson model can also be applied here, but are not shown. The logistic model can be extended using multiple covariates or using a combination of both covariates (continuous predictor variables like age or parasite density) and factor effects (categorical predictor variables like gender or infection status). Two logistic regression graphs are shown here (Fig 5f,g), but they aren't commonly used in publications.

Basic Kaplan-Meier survival curves (Fig. 5h) can be fit and plotted using the `survfit()` function from the survival package library. First, a "survival object" must be created using the `Surv()` command, which essentially adds a special marking to identify each data point as censored or uncensored. After the Kaplan-Meier curves are fit using `survfit()`, multiple Kaplan-Meier curves can be compared using the `survdif()` function. A more complicated Cox regression can be fit using `coxph()`.

Categorical Response Models in GraphPad Prism—GraphPad Prism has a contingency table format (Supplementary File 8) that is mostly limited to basic contingency table analyses using Fisher's exact test for 2×2 tables or Pearson's chi-squared test for 2×2 and larger tables (https://www.graphpad.com/guides/prism/7/statistics/index.htm?stat_categorical_outcomes.htm). Prism cannot compute more complicated generalized linear models, like Poisson family models or binomial family logistic regressions. Prism also has a survival table format that can be used compute Kaplan-Meier survival curves, but not Cox regression or more advanced survival models (https://www.graphpad.com/guides/prism/7/statistics/index.htm?survival_curves.htm).

Basic Protocol 5. Statistical Design of Experiments

Some biologists might assume that statisticians only analyze data, but statistics can also be used to help plan new experiments, before any data has been collected. This is the field of statistical design, or design of experiments (DOE). Far and away the most common application of statistical DOE in immunology is the estimation of minimum sample size required for human and animal studies, but proper DOE can also help properly randomize data and minimize problems like reagent or technician batch effects.

Power and sample size estimates exploit the known relationships among approximately 5 different parameters:

- Sample size: n
- Difference between groups: δ
- Standard deviation: σ
- Type I error rate: α
- $1 - \text{Type II error rate}$: power

These five parameters are related to one another through the relationship between the null and alternative distributions used to calculate a hypothesis test (Fig. 6a). The δ parameter describes the distance between the null and alternative distributions (Fig. 6b). The σ and n parameters describe the widths of the null and alternative distributions (Fig. 6c). The Type I and Type II error rates are determined by the critical value (e.g. $p = 0.05$) used to distinguish significant and nonsignificant test results (Fig. 6a). Define or estimate four of these parameters to estimate the fifth unknown experiment parameter.

Typically, you want to predict minimum sample size or statistical power, but any of the 5 parameters could be predicted. Often, researchers might compute a range of multiple sample size (Fig. 6d) or power predictions (Fig. 6e) or δ predictions (Fig. 6f) for a series of estimated standard deviations or δ values. Often such a series of sample size (Fig. 6g) or power (Fig. 6h) or δ (Fig. 6i) estimates might be displayed in a graph that shows the relationships between the five parameters. Power and sample size estimates for t-tests, linear regressions and even ANOVA are relatively straight forward with only minor variations on the 5-parameter calculations described above. However, power and sample size estimates can be calculated for almost any statistical model, including some very complicated analyses, like gene expression microarray (Fig. 6j) or RNA-seq experiments.

Statistical design is a deep field that covers much more than simple power and sample size estimates. Many basic experimental designs, like randomized complete block designs or split-plot designs describe how experimental conditions should be translated into statistical analyses, often complicated ANOVA models. Properly modeling an experimental design can help remove nuisance factors from the residual error of your model, making statistical tests more significant. Immunologists may work on clinical trials, which have their own special experimental designs like crossover designs, case-control studies, prospective and

longitudinal studies. These designs both ensure good science and help determine how the resulting data should be analyzed.

More importantly, considering the design of an experiment prior to collecting data can help avoid critical design mistakes, like collinear or confounded effects. For example, you could suppose you were to run microarray experiments on serum samples collected from patients before malaria transmission season and after malaria transmission season. However, suppose that the pre-season samples were all analyzed by one batch of microarrays, while the post-season samples were all analyzed on a different batch of arrays. In this case, you could never know whether the differences in the pre- and post-samples were due to the effects of transmission season or due to the differences between the microarray batches. It would have been preferable to purchase the two batches of microarrays necessary to run the experiment beforehand, then randomly assign half of the pre-season samples to slides from each batch, and randomly assign half of the post-season samples to slides from each batch so the batch and season effects could remain independent.

Two special types of experimental designs that often require statistical planning are screening designs and response surface designs. Screening design experiments are used when researchers would like to investigate multiple factors that influence a process in a single model, rather than investigating one factor at a time. For example, a biologist might be struggling to isolate a particular type of cell using a flow cytometry protocol, so they might investigate 10 factors that could possibly influence the process and lead to low yields. Studying multiple factors at the same time in an efficiently designed experiment can save lots of time, money and resources over exploring one factor at a time. Common screening designs include fractional factorial designs and Plackett-Burman designs, both of which can be generated using statistical software. Screening designs are typically analyzed using ANOVA and regression techniques, though special tests and graphics like half-normal plots (Fig. 6k) might be used to identify the most significant factors in the screening experiment.

Screening design experiments are often followed up by a second response surface design experiment, using a smaller set of the most significant factors from the screening design to find the optimal combination of factor values to maximize an assay yield or minimize an experimental cost. The response surface design uses more combinations of factor values than a screening design to get a better picture of the complex relationships among variables and find their best combinations. Typical response surface designs include central composite designs and Box-Behnken designs. The final result of a response surface experiment is usually a 2D contour plot (Fig. 6l) or a 3D surface plot, which is used to find the maximum or minimum response.

Examples

Statistical Design of Experiments in R—There are dozens of power and sample size calculations available in R (Supplementary File 9), but they are scattered among dozens of different package libraries: `samplesize`, `ssize.fdr`, `pwr`, `powerpkg`, `powerAnalysis`, `powerbydesign`, `Sample.Size`, `samplesize4surveys`, etc. There are also basic power and sample size functions in the default `stats` library. Often specific power and sample size libraries will address specific types of statistical tests and experimental designs, while other

libraries may be more general. Several examples are shown using the functions `power.t.test()` and `power.anova.test()` in Supplementary file 9.

In the first examples, the `power.t.test()` is used to calculate 3 single estimates for sample size, power and delta. Then `power.t.test()` is used to compute 9 estimates of sample size (Fig. 6d), power (Fig. 6e) and delta (Fig. 6f) to plot some basic power and sample size curves. Likewise, in the second series of examples, the `power.anova.test()` function is used to calculate 3 single estimates of sample size (Fig. 6g), power (Fig. 6h) and delta (Fig. 6i), before multiple sequences of estimates are calculated and plotted. Finally, `ssize.twoSampVary()` from the `ssize.fdr` package library is used to calculate power estimates for a typical gene expression microarray analysis (Fig. 6j).

Several examples are shown generating experimental designs for new experiments. The function `get.plan()` from the `crossdes` library is used to generate variations on a crossover or carryover design through an interactive series of 2–3 menu steps. The function `find.BIB()` from `crossdes` is used to generate balanced incomplete block designs, which occur when a blocking factor cannot accommodate all the treatments from another factor. For example, you might have a longitudinal study with 4 time points, but your experiment uses a 96-well plate design that can only accommodate 3 samples at a time. The functions `FrF2()` and `pb()` from the `FrF2` library are used to generate fractional factorial and Plackett-Burman screening designs, respectively. Generating some random data for the Plackett-Burman design, we can see a quick example of a half-normal plot, which might be used to identify the most important factors from a screening design (Fig. 6k).

The functions `ccd()` and `bbd()` from the `rsm` library are used to generate central composite and Box-Behnken designs, respectively, for response surface design experiments. Again, generating some random data for the Box-Behnken design, we can create a contour plot (Fig. 6l) to explore the actual response surface and identify the predictor value combinations that result in a maximum or minimum response, depending on whether you would like to maximize yield or minimize costs, for example.

Statistical Design of Experiments with JMP—The software JMP (www.jmp.com) can be also be used to compute power and sample size estimates or generate special experimental designs, like screening or response surface designs (http://www.jmp.com/support/help/Design_of_Experiments_Guide.shtml). Power and sample size estimates can be computed with simple point-and-click menus for t-tests (one- and two-sample means), ANOVA (k=sample means) and other analyses (http://www.jmp.com/support/help/Prospective_Sample_Size_and_Power.shtml#65477). Screening designs like fractional factorials and Plackett-Burman can be generated through the Screening Platform (http://www.jmp.com/support/help/Screening_Designs.shtml#112764), while response surface designs like central composite and Box-Behnken designs can be generated from the Response Surface Platform (http://www.jmp.com/support/help/Response_Surface_Designs.shtml#67894).

Basic Protocol 6. Advanced Methods: Resampling, High-throughput, Bayesian Statistics and Data Mining

The protocols presented above should cover most of the statistical analyses you will encounter in a typical immunology paper or your own immunology experiments. However, there are MANY more advanced statistical methods and you may encounter some of them in your reading or your own research. There's no way to cover every possible statistical method in one short paper, but it may help to review a few major types of analyses that you may encounter in more complicated immunological papers, so you can be familiar with their basic concepts and why they might be used in the given situation.

Sometimes the choice between parametric t-tests and ANOVA or typical nonparametric MW-tests and KW-tests is difficult or even impossible. You might encounter a data set that doesn't meet the assumptions of either the t-test or even the minimal assumptions of the MW-test. One possible example is a comparison of a placebo group with high bacterial loads and an antibiotic treated group with exactly zero bacterial load in all samples (Fig. 7a). In this situation, it should be obvious to all readers that the placebo and treated groups are different, even without a statistical test. Still, some reviewers or journals may require a p-value, even in situations like this where groups are so clearly different. For practical purposes, both the t-test and the MW-test would also yield the "correct" answer, but the experiment technically violates the assumptions of both tests and therefore neither test "should" be used in the strictest theoretical sense. If you wanted to be very thorough and use a test where no assumptions had been violated, you could try a resampling test (Good 2006).

In a regular t-test or ANOVA or linear regression, the central limit theorem tells us that the distribution test statistics, t^* or F^* , from repeated samples of the same size N should converge to a known distribution, like the t-distribution or F-distribution. The basic idea behind the resampling test is that the sample itself can be used as a model for the sampling distribution whenever the theoretical sampling distribution isn't known. This is achieved by repeatedly drawing new "resamples" from the original sample hundreds or thousands of times, using techniques like permutation or bootstrap resampling. These resamples can then be used to compute a statistical test or estimate a confidence interval (Fig. 7b,c,d). You can use resampling to compute a "true" nonparametric test of the median, since the median has no standard deviation or standard error to compute a regular t-test. Resampling also works well in many "cutting edge" applications, because the resampling idea is so easily generalized to new problems.

One downside to resampling tests is that precise p-value calculations are cannot be replicated, even for one specific sample. With a regular t-test or regression, you can calculate the same test on the same sample 1000 times and you will always get the same answer. With resampling tests, if you calculate the same resampling test on the same data set 1000 times, you will probably get 1000 slightly different p-values. The p-values should be consistent. If your first resampling test yields a p-value of 0.00043, then your second resampling test will likely also be significant at about the 0.000x level. However, your precise p-values will vary slightly every time you re-compute the test. It is worth noting that resampling tests are sometimes called "exact" tests, not because they are easily replicable, but because their p-

values are equally robust at all sample sizes, unlike regular statistical tests that rely on the assumptions of large sample sizes.

Permutation resampling typically works using permutations to randomly swap the data labels in a problem like a two-sample t-test or ANOVA. By swapping group labels, permutation resampling assumes no differences between the groups and effectively models the null distribution, therefore it is best for applications where you want to compute a p-value. Bootstrap resampling works by drawing random sample with replacement from the existing sample groups without swapping group labels, which generates unique sampling distributions for each group and effectively models the alternative distribution and makes it better suited to confidence interval problems, though it can be used to compute p-values too.

Another method that may come up in your reading is the idea of Bayesian statistics (Gelman and Carlin 2013). All the methods presented so far in this manuscript have been “frequentist” statistics methods, which means p-values and confidence intervals are estimated by the how frequently certain types of samples are likely to occur under the null hypothesis. Bayesian statistics are a completely different statistical framework, based on conditional probabilities calculated using Bayes’ Rule. Bayesian statistics assume that the parameters of your statistical model are random variables themselves, and they can be sampled from known statistical distributions called “prior distributions”. These prior distributions represent our assumptions about the model parameters, which can either be strong assumptions (i.e. informative priors) or weak assumptions (i.e. noninformative priors). Inferences from the Bayesian model are made by drawing random samples from a posterior distribution, which is a product of the actual data and the prior distribution. These random samples can be used to estimate a distribution of the possible parameter values (Fig 7e,f). Two common complaints against Bayesian statistics are that prior distributions can be too informative, potentially biasing statistical results, and that the results may not be precisely reproducible, because they are based on random sampling just like the resampling methods. However, Bayesian tests have desirably properties too, and they can be extended to many cutting-edge problems that are not well served by traditional, frequentist statistics.

Over the last 30 years, advancements in computing power and the introduction of high-throughput biology experiments like microarrays and next generation sequencing have brought new statistical challenges, as researchers try to make sense out of huge amounts of biological data (Parmigiani *et al.* 2003). One of the main challenges is that computing such a very large number of statistical tests fundamentally changes how we handle adjustments for multiple comparisons and the estimation of variances. When we compute a “small” number of statistical tests (e.g. 3–100 tests) in an ANOVA or other model, we use adjustments that assume the entire “family” of multiple tests should have zero false positives. This is sometimes called a “family-wise error rate” (FWER) adjustment. When computing thousands or millions of statistical tests, it becomes unreasonable to assume we would find zero false positives, so instead we seek to limit the number of false positives to a reasonable rate. This is sometimes called a false discover rate (FDR) adjustment. High throughput microarray and sequencing experiments often use FDR multiple comparison adjustments with some kind of “error shrinkage” method, like empirical Bayes shrinkage (e.g. limma, edgeR) or permutation methods (e.g. SAM, deSEQ), to better estimate the variances of

individual genes or features. Many specialized tests and plots are used to ensure that high throughput data are properly processed and normalized and that results are properly interpreted. For example, a principle components analysis (PCA) plot (Fig. 7g) might be used to look for nuisance effects in the microarray processing (e.g. batch effects from chips made at two different time points) or a “volcano” plot might examine the relationship between significance and fold change for gene expression results (Fig. 7h).

High throughput data or large “meta analyses” of thousands of data sets stored on databases can also be analyzed using machine learning or data mining techniques (Hastie and Tibshirani 2016). These methods differ from the tests discussed throughout the rest of this manuscript, because most are only used to classify or explore large datasets without computing p-values or hypothesis tests. Instead these methods are usually evaluated based on their accuracy and consistency. Statisticians and computer scientists often group data mining techniques into two categories: unsupervised learning and supervised learning (Hastie and Tibshirani 2016).

Unsupervised learning methods explore data sets using only the dependent, response variables from the experiment, typically high-throughput measurements like gene expression data, biometric data (e.g. finger prints, retinal images, facial features), or survey response data. The most commonly used unsupervised learning models include PCA (Fig. 7g) and hierarchical clustering. These methods group samples together purely based on the similarity or proximity of their responses, e.g. grouping tumor samples according to the expression of their genes, as detected by microarray or RNA-seq. PCA is a mathematical transformation of the response data into a new coordinate system of independent, or “orthogonal” components. You can picture a full set of expression data as representing a large 20,000-dimensional space, where each dimension is a different gene and many of the genes are very similar or colinear to each other. If you could visualize such a 20,000-dimensional object, it would have partially overlapping dimensions coming out of all directions. The PCA transformation uses eigenvalue decomposition to ensure that each new dimension is independent and perpendicular to all other dimensions. Each new dimension represents a linear combination of the original gene expression dimensions. More importantly, the new dimensions are ordered by the magnitude of the effect they have on the variability of the data. So, the first principal component PC1 controls more of the variation in the data than the second component, PC2. This is helpful, because most of the variability in the data can be captured in the first 2 or 3 principal components, simplifying the creation of PCA plots (Fig. 7g). The resulting PCA plot will typically show the data clumped together in 1 or more groups. Even though the PCA calculations do not use any group labels, it is often helpful for researchers to plot the PCA data using color or symbol types to represent previously known groupings, like male of female subjects or infected vs uninfected subjects, to see if the data clusters together in those categories in the PCA plot (Fig. 7g).

Supervised learning methods use known grouping variables to define their classification models. Technically, the logistic regression model presented in Basic Protocol 4 is a supervised learning method, but more typically data miners use advanced methods like K-nearest neighbors (KNN) or random forests (RF). These methods can be used to classify tumor samples as malignant or benign using gene expression data, for example. Unlike a

logistic regression, which might evaluate the significance of 1–2 covariates in classifying a sample, these machine learning techniques utilize dozens or hundreds of predictor variables and offer no p-values to assess the importance of individual genes or features, instead focusing only on model accuracy and consistency. Each method has its own mathematical and conceptual basis for classifying samples, for example KNN methods classify each sample using the identities of the nearest, or most similar, samples in the data set, while a random forest model is built up from multiple classification and regression “trees”, which attempt to classify samples using a handful of the response variables (e.g. sample is infected if $\log_2 \text{STAT6 expression} > 6.8$). One key issue in supervised learning models is the idea of parameter training. For example, the K nearest neighbors model can look at $K = 3$ neighbors or $K = 7$ neighbors when classifying a sample, and the choice between $K = 3$ or $K = 7$ could affect the classification. Maybe a sample would be classified as infected with $K = 3$ neighbors, but classified as uninfected with $K = 7$ neighbors. Consistency is another key issue in data mining, since results can easily be “overfit” to one specific data set, so they won’t generalize to other data sets. Statisticians generally try to use “cross-validation” to chop a large data set up into multiple pieces that might act like independent data sets to evaluate the reproducibility of the data mining procedure. For example, in 6-fold cross-validation, the data might be chopped into 6 equal sized pieces, so the data could be repeatedly “trained” on 5/6 pieces, then tested on the 6th piece. This would yield 6 different independent estimates of the accuracy of the classification model, so you could compare the average performance of the model among each of the six folds and you could calculate the cross-validation error among the six folds. You could then use these cross-validated evaluations (Fig. 7i) to choose among different parameter values for a model like KNN or to choose among entirely different types of models (e.g. KNN and RF)

Examples

Advanced Methods in R—Although there are many R packages (Supplementary File 10) that use resampling tests like permutation tests and bootstrap tests, and there are even a few basic resampling tests to handle simple questions like a two sample t-test of the median, many resampling tests will need to be created using custom R functions like those written in `permtest.R`. The example (Fig. 7a) shows a data set that doesn’t meet the assumptions of either Student’s t-test (non-normal data and unequal variances) or the Mann-Whitney test (data from two completely different types of distributions). Supplementary File 11 shows two types of resampling tests, one using permutations of the difference in sample medians (Fig. 7b,c) and the other using bootstrap resamples of differences in the 40th percentile (Fig. 7d). Keep in mind that other types of permutation or bootstrap tests can be developed for many other types of experiments.

The permutation test (Fig. 7b,c) scrambles the group labels, which leaves most of the resampled differences in medians (tan histogram bars) in a normal distribution centered around 0 with a few random samples approaching the actual difference in sample medians from the real data (vertical red line). Essentially, the permutation resampling method estimates the null distribution using the assumption that the two groups are the same (i.e. the labels don’t matter). If the actual observed difference in medians (red line) falls outside the middle 95% of the resampled data, then permutation test is statistically significant. If the

actual differences in sample and resample medians are used, then you have a one-sided statistical test (Fig. 7b), but if the absolute values of the differences in sample and resample medians are used, then you have a two-sided test (Fig. 7c).

The bootstrap resampling method (Fig. 7d) draws repeated samples with replacement from each sample group with no label switching, which means the resamples look quite similar to the actual data from each group. Therefore, differences in the resampled 40th percentiles will appear in a normal distribution (tan histogram bars) around the actual difference in 40th percentiles from the real data (vertical red line). Essentially, the bootstrap resampling method estimates the alternative distribution, which you can compare against a fixed null hypothesis value (vertical blue line). If the null hypothesis value falls outside the middle 95% of the resampled data, then the bootstrap test is statistically significant.

Some simple examples of Bayesian t-tests and regressions from the `BayesFactor` library are shown in the second group of examples. The function `ttestBF()` computes a Bayesian t-test, while `lmBF()` can compute any Bayesian linear model, but specifically fits a linear regression in this example. A connected line plot of the Bayesian t-test coefficients over 1000 posterior samples (Fig. 7e) and a density plot of the Bayesian regression slope coefficient (Fig. 7f) are shown to illustrate the posterior sampling process. These are just two ways of showing the same basic concept. The connected line plot (Fig. 7e) shows if the posterior distribution samples are somehow correlated over time, which would be bad, while the density plot (Fig. 7f) merely shows the final distribution of all the posterior sample draws for a better picture of the mean and SD of the regression slope parameter sampled from the posterior distribution.

In all the rest of this manuscript, we have used randomly generated data to demonstrate the tests and plots of each section. In the third example of this section, we can use real microarray data from the `bcellViper` library. This microarray analysis is greatly simplified in my script, but the first steps deal with extracting the necessary data sets. Generally, gene expression experiments require the expression data table itself (i.e. a large table with genes on the rows and samples on the columns), some phenotype data describing the samples and some annotation data describing the genes. Usually, you would also walk through multiple steps of quality control checks, transformations and normalizations, but those steps are skipped here with the pre-packaged data. The actual statistical tests are computed in multiple steps, starting with the specification of the experimental design matrix, fitting the statistical model with `lmFit()`, specifying the model contrasts, and saving the model results with `topTable()`. A PCA plot (Fig 7g) is used to determine if any batch effects or other nuisances might obscure the differences among treatment groups. A heatmap (not shown in figure) is generated using `heatmap.2()`, revealing differences in expression patterns among the samples (columns) and genes (rows). Finally, a volcano plot (Fig 7h) is constructed from the saved p-values and log fold changes to show the proportion of genes with significant p-values and substantive fold changes.

Similar to these other advanced methods, you can find many R packages for machine learning and other data mining techniques, but most require some customization. Supplementary File 12 contains the actual cross-validation and datamining steps, but very

briefly it uses $K = 4$ fold cross-validation to fit KNN and RF models to the sample data. In other words, the large data set is chopped into 4 pieces, then sequentially fit to 3 of those pieces and tested with the 4th piece to calculate prediction accuracy and Cohen's Kappa coefficient, which evaluate the performance of each classifier. Each time the models are fit and tested to unique chunks of the data is called a cross-validation "fold". The graph (Fig 7i) then plots the mean accuracy, mean Kappa and their standard deviations from among the 4 folds of the cross-validation process, providing an estimate of the performance of the model among different subsets of the original sample. Ideally, you would like to choose a model that has both high accuracy and high Kappa, with small accuracy and Kappa errors. The accuracy and Kappa represent the model performance, while the size of the errors represent how consistently the model can be reproduced. In this example, the random forests (RF) model has both higher average accuracy and Kappa with smaller accuracy and Kappa standard deviations, making random forests the clear "winner" among the models tested.

COMMENTARY

Critical Parameters

Almost all statistical tests rely upon specific assumptions about the qualities of your data. Are the data normally distributed? Are the variances equal? Are the risks or hazards in a survival model proportional at all time intervals? Learn the assumptions of each model you use and check to see if your data meets those assumptions.

Troubleshooting

First and foremost, any immunologist or biologist should try to find a statistician they can consult and collaborate with at their lab, agency, institute, school, company, etc. Statistics is a large discipline and there really is no substitute for education, experience and expertise. When it comes to specific statistics software issues, there are lots of resources available to you. Many authors have written books and guides about the R Project software (de Vries and Meys 2015, Teetor 2011, Davies 2016, Lander 2017), and there are plenty of free resources available on the web, including the official R-help mailing lists (<https://www.r-project.org/mail.html>), the support website R-bloggers (<https://www.r-bloggers.com>), StackOverflow (<https://stackoverflow.com/questions/tagged/r>) and hundreds of academic websites from university statistics departments across the globe. Commercial statistics software products like GraphPad Prism, JMP Statistical Discovery and others will usually maintain their own support websites and support staff, typically available by email or phone.

Anticipated Results

Throughout these protocols, I have tried to provide guidance and examples of how to make useful graphics and statistical tests to evaluate your biological data. Try to remember that statistics isn't just the hunt for a significant p-value. Statistical tests are additional pieces of evidence to support or possibly refute your biological hypotheses. It's better to use the correct statistical test and accept its results, than to report a "significant" result from the wrong statistical test. Computing a large number of potentially inappropriate statistical tests until you find any result with a significant p-value is a terrible and fraudulent practice,

sometimes refer to as “p-hacking”. Try to avoid these bad practices and please consult a statistician if you are unsure about your results.

Time Considerations

With modern computers, most statistical analyses and graphs are produced instantaneously, as soon as you submit your code or click your way through the appropriate menus. However, some of the more advanced methods presented here that rely on resampling, Bayesian MCMC sampling or data mining cross-validation methods might take longer to compute. Computation times vary from a few seconds to hours or even days. Often the lengthiest computations can be distributed among multiple computing nodes on a high-performance computing cluster, like NIH BioWulf (<https://hpc.nih.gov/systems/>) or the Amazon Cloud (<https://aws.amazon.com/>) to speed things up.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

This manuscript was supported by the Intramural Research Program of the National Institute of Allergy and Infectious Diseases Department of Intramural Research, National Institutes of Health. Jeff Skinner would like to thank David Margulies, Ann Boyle, Peter Crompton, Christine Hopp, Anton Goetz, Rajan Guha, Prasida Holla, Valeria Lodde, Shanping Li and Andrew Orcutt for their helpful review and comments.

LITERATURE CITED

- Koenker RW, Bassett GW. Regression quantiles. *Econometrica*. 1978; 46:33–50.
- Koenker RW. Quantile Regression. Cambridge U. Press; 2005.
- Hettmansperger TP, McKean JW. Kendall’s Library of Statistics. 1998. Robust nonparametric statistical methods.
- Motulsky HJ. Statistics Guide. GraphPad Software; San Diego CA: 2018.
- Student. The probable error of a mean. *Biometrika*. 1908; 6(1):1–25.
- Utts JM, Heckard RF. Mind on Statistics. 5. Brooks Cole; 2014. This is the long-running introductory, statistics for science majors (200-level) textbook used at the University of California Irvine (UCI) and Penn State University (PSU). It’s a great introductory text, because it was one of the first textbooks written for students who would be learning statistics with point-and-click software. As a graduate student, I taught undergraduates using the Second Edition version of this text and all that material remains pretty relevant today. You should be able to find a cheap used copy that will serve you well for many years
- Wilcoxon F. Individual comparisons by ranking methods. *Biometrics Bulletin*. 1945; 1(6):80–83.
- Good P. Resampling Methods. 3. Birkhauser; 2006.
- Parmigiani G, Garrett ES, Irizarry RA, Zeger SL. The Analysis of Gene Expression Data. Springer; 2003.
- Gelman A, Carlin JB. Bayesian Data Analysis. 3. Chapman & Hall; 2013. This is a graduate level textbook, and probably one of the best resources on Bayesian statistics available
- Hastie T, Tibshirani R. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2. 2016. This is another graduate level textbook, but it’s a great reference aimed at a wide audience of statisticians, computer scientists, biologists and others
- Kutner MH, Nachtsheim CJ, Neter J, Li W. Applied Linear Statistical Models. 5. McGraw-Hill; 2004. This is a graduate-level textbook on linear models, and a great resource if you want a deeper

understanding of linear regression and ANOVA, how their calculated, model diagnostic plots and tests, etc. Despite the title, this book even has a decent chapter on nonlinear regression methods

Agresti A. Categorical Data Analysis. 3. Wiley; 2012. This is another graduate level textbook, and the definitive resource for generalized linear models and other categorical response models

de Vries A, Meys J. For Dummies. 2. 2015. R for Dummies.

Teetor P. R Cookbook: Recipes for Data Analysis, Statistics, and Graphics. O'Reilly Media; 2011.

Davies TM. The Book of R: A First Course in Programming and Statistics. No Starch Press; 2016.

Lander JP. R for Everyone: Advanced Analytics and Graphics. 2. Addison-Wesley Professional; 2017.

Fig. 1.

Flow-chart illustrating general approaches to use of statistics in immunology. Starting at the upper left, the first table describes how a researcher might choose an appropriate graph (Protocol 1). Moving to the right, the second table shows how a researcher might choose among basic statistical tests like t-tests and ANOVA (Protocol 2), correlations and regression (Protocol 3), or contingency tables and generalized linear models (Protocol 4). When those typical statistical analyses are inadequate, the graphs in the lower right-hand corner show some examples of advanced methods (Protocol 6). The lower left-hand corner shows some methods for statistical design of experiments (Protocol 5), which is used to plan new experiments and choose analyses for existing experiments. The flow between topics shows how researchers will often move from looking at data to choosing new analyses, and how design of experiments can often circle the process back around to looking at newly collected data.

Fig.2.

Best Practices for Statistical Graphs. (A) barplot with standard error of the mean (SEM) error whiskers for four groups. (B) “beeswarm” style scatterplot with mean lines and standard deviation (SD) error whiskers showing the exact same four samples as Fig.2a. (C) barplot with SEM error whiskers for four new samples of data. (D) beeswarm plots with overlaid Tukey boxplots (blue) and mean lines (red). (E) pie chart displaying the percentages of four categorical groups. (F) barplot showing the counts within four categorical groups. (G) beeswarm plots with mean lines and unnecessary color and symbol shape changes among groups for same data in Fig.2c,d. (H) overly complicated 3D barplot with several groups obscured. (I) (J) (K) (L) four random samples of the same data type, but one plot (K) inconsistently uses a barplot figure instead of the established scatterplot.

Fig.3.

t-tests, ANOVA and basic nonparametric tests. (A) “beeswarm” scatterplot with mean lines and standard deviation whiskers for an unpaired Student’s t-test. (B) beeswarm scatterplot with Tukey boxplot for Mann-Whitney test. (C) paired data plot for paired Student’s t-test or paired Wilcoxon Sign Rank test. (D) three normal distributions with more variation within groups than among groups. (E) three normal distributions with more variation among groups than within groups. (F) beeswarm scatterplot with mean lines and standard deviation whiskers for one-way ANOVA with Tukey post hoc tests. (G) beeswarm scatterplot with Tukey boxplot for Kruskal-Wallis with Dunn’s post hoc tests. (H) matched data plot for repeated measures one-way ANOVA. (I) two-color matched data plot for repeated measures two-way ANOVA of PCR– (red) and PCR+ (blue) sample groups.

Fig.4. Linear and Nonlinear Regression. (A) weak correlation. (B) mild correlation. (C) strong correlation. (D) nonlinear correlation. (E) simple linear regression. (F) residual plot to assess model fit and assumptions. (G) effect of a high leverage point on a linear regression. (H) leverage plot to identify high leverage point (I) quantile regression for tau = 10th percentile. (J) Analysis of Covariance (ANCOVA) plot with 95% confidence bands. (K) quadratic polynomial regression (L) logistic dose-response curve fit using nonlinear regression.

Fig.5.

Categorical models and Survival. (A) two-color bar chart for 2×2 contingency table analyzed with Fisher's exact test and Pearson's chi-squared test of independence. (B) three-color bar chart for 3×4 contingency table. (C) two-color bar chart for 2×2 contingency table analyzed with McNemar's test for paired data. (D) two-color multipanel bar chart for Cochran-Mantel-Haenszel test of two 2×2 contingency tables. (E) two-color multipanel bar chart for multifactor contingency table analyzed with generalized linear model. (F) simple logistic regression curve. (G) logistic regression model with added factor effect (H) Kaplan-Meier survival curves with 95% confidence bands for two groups. (I) Kaplan-Meier survival curves with 95% confidence bands and color gradient to account for Cox regression covariate.

Fig.6. Design of Experiments. (A) null and alternative distributions. (B) null and alternative distributions with “Delta” parameter marked (C) null and alternative distributions with $\text{Sigma} = \text{SD}/\sqrt{n}$ marked. (D) relationship between increasing delta and minimum sample size in a power and sample size estimation. (E) relationship between sample size and power. (F) relationship between power and delta. (G) relationship between delta and sample size for 3 different SD estimates. (H) relationship between sample size and power for 3 SD estimates. (I) relationship between power and delta for 3 SD estimates. (J) power and sample size estimates for gene expression microarray data. (K) Half-normal plot for analysis of screening design experiment. (L) contour plot of regression surface for response surface optimization experiment.

Fig.7.

Advanced Methods. (A) beeswarm plot of data that violates the assumptions of both Student's T-test and Mann-Whitney-Wilcoxon nonparametric tests. (B) histogram of permutation resampled data for one-sided p-value calculation. (C) histogram of absolute value permutation resampled data for two-sided p-value calculation. (D) histogram of bootstrap resampled data. (E) Bayesian T-test coefficients plotted over sample iteration (F) density plot of Bayesian regression slope (G) Principal Components Analysis (PCA) plot of microarray data (H) "volcano" plot from gene expression analysis of microarray data (I) plot of data mining accuracy and Cohen's Kappa coefficient with cross-validation error measurements.

Table 1

Types of Statistical Errors.

		Population Means	
		Different	Not Different
Statistical Test	Significant	OK	Type I error False Positive
	Non-significant	Type II error False Negative	OK

Table 2

Example of a 2×2 contingency table.

		Status		
		Susceptible	Immune	
PCR	Negative	11	23	34
	Positive	19	6	25
		30	29	59

Table 3

Example of a 3×4 contingency table.

	Fever			
	No Fever	Mild Fever	High Fever	Dangerous
A	20	19	33	36
B	16	8	42	38
C	35	27	11	6

Table 4

Example of a paired 2×2 contingency table for McNemar's Test

		Infection		
		Uninfected	Infected	
Time	Pre	41	24	65
	Post	17	38	55
		58	62	120

Table 5

Comparison of two 2×2 contingency tables using Cochran-Mantel-Haenzel Test.

		Region			
		Africa		US	
		Status			
		Susceptible	Immune	Susceptible	Immune
PCR	Neg	11	23	21	5
	Pos	19	6	9	27

Table 6

Tall format table for log-linear model.

				Count
Susceptible	Neg	Africa	Infant	9
Immune	Neg	Africa	Infant	6
Susceptible	Pos	Africa	Infant	4
Immune	Pos	Africa	Infant	2
Susceptible	Neg	US	Infant	2
Immune	Neg	US	Infant	2
Susceptible	Pos	US	Infant	4
Immune	Pos	US	Infant	0
Susceptible	Neg	Africa	Child	15
Immune	Neg	Africa	Child	15
Susceptible	Pos	Africa	Child	14
Immune	Pos	Africa	Child	10
Susceptible	Neg	US	Child	10
Immune	Neg	US	Child	16
Susceptible	Pos	US	Child	19
Immune	Pos	US	Child	15
Susceptible	Neg	Africa	Adult	71
Immune	Neg	Africa	Adult	93
Susceptible	Pos	Africa	Adult	75
Immune	Pos	Africa	Adult	77
Susceptible	Neg	US	Adult	51
Immune	Neg	US	Adult	94
Susceptible	Pos	US	Adult	52
Immune	Pos	US	Adult	301