1 Installation

To use the Label Helper, first you need to install the **node.js** framework, along with the package manager **npm**. For that, go to `https://nodejs.org` and follow the instructions for your system. You will also need a browser (any modern browser will do) and git installed on your system (most Linux distros have it preinstalled)

Clone the repo by entering the command
git clone https://github.com/RedTachyon/label-helper into your terminal. Then, navigate into the newly created directory (**cd label-helper**) and run **npm install**. That will install all the necessary dependencies.

2 Usage

To run the app, enter the command **node app.js** in the root directory. Your default browser should automatically open the app - if not, navigate to **localhost:1337**.
After a short pause (which could take up to several seconds on weaker machines) you should see a graph displaying the data and several buttons and indicators below it.

When the app is running, you can click anywhere on the graph to mark the chosen point - its time value will be automatically saved to the **out/points.txt** file. At any moment you can also press **u** or the button labelled as **Undo (u) (!)** to delete your last click. You can move forward and backward in the data by pressing right/left arrow. By default, you're shown the first 16 seconds, and moving forward moves the window by 15 seconds (so that there's always a 1 second overlap).

There's also a number of buttons for convenience and for being able to differentiate between various types of points to be marked. By pressing the **t** button (or, as always, the appropriately labelled button below the graph) you can change which line you're marking - the red one or the blue one (color-coded for convenience). By pressing the **m** button you can toggle between marking jumps, noisy regions or hills. Each jump is a single point, but both noises and hills require a start and an end point. The app generally toggles between start and end automatically, but in case the need arises, you can change the current state with the **j** button. The current state is also displayed below the buttons.

In case you need to inspect only a single temperature line, you can toggle the visibility of the blue/red line by pressing **1**/**2** respectively.
There's also a possibility to change the number of points displayed on the graph at any given moment (**Current window length**), and the distance

traversed with each click of a left or right arrow (**Current window step**). There's also the option to manually move to a chosen location by using the **Move to location** input box.

Finally, there's the **SAVE** button, not bound to any physical buttons. All it does is creating a backup of all currently saved points to a file named **points.txtbackup⟨n⟩** where ⟨n⟩ stands for the index of the backup.

Feel free to experiment with all the settings to get a feel of how they work individually and interact with each other.