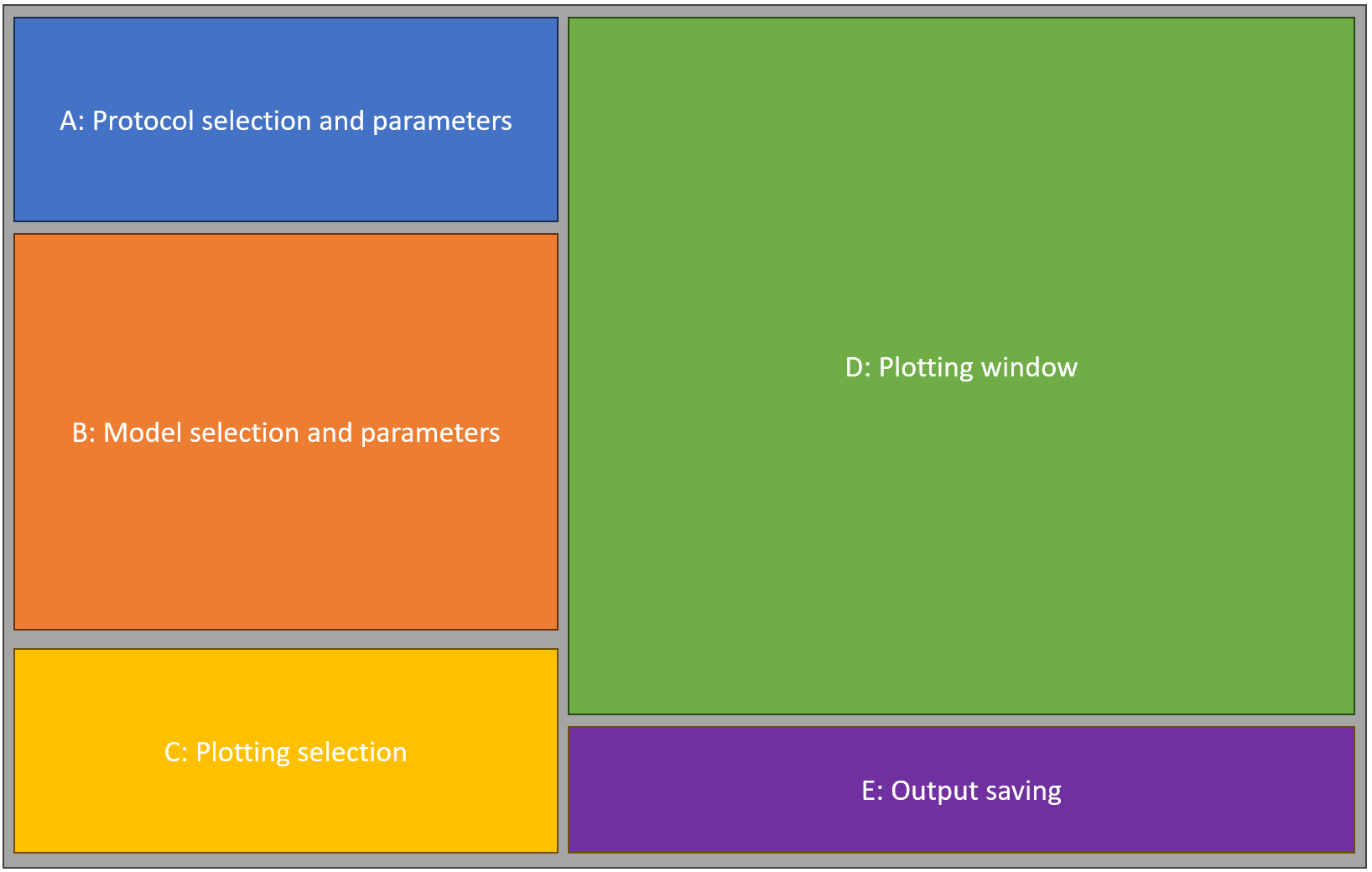
# Layout design



# A: Protocol selection and parameters

Here the user chooses which stimulation protocol to apply.

* **Protocol:** drop down menu (top left). Depending on the value selected, other elements will be displayed and activated.
  + **‘Regular stimulation’:** Standard, fixed-rate pacing. Default will be 1 Hz, 10 beats, showing the last one.
    - **Basic cycle length (ms):** An input textbox with a label. It would be neat to have that followed by a textbox that would show the corresponding rate in beats per minute. Default = 1000.
    - **Number of beats:** How many beats are being simulated. Default = 10.
    - **Showing last X beats:** How many beats are shown in the plotting window. Default = 1. (*Question: would this better fit in section C?*)
  + **‘Early afterdepolarization’:** This would be really just a shortcut, having the same controls as regular stimulation, just having different default values. Namely, basic cycle length would be 4000, default Cao (extracellular Ca) would be 2 mM, and default IKr\_Multiplier would be 0.15. This can reuse the control elements of the “Regular stimulation”.
  + **‘S1-S2 restitution’:** This will simulate a typical S1S2 restitution. It will have a different section C than the previous two protocols.
    - **S1 cycle length (ms):** Input text box; the pacing rate for steady-state pacing. Default is 1000.
    - **Number of S1 beats:** Input text box. Default is 10.
    - **S2 intervals (comma separated):** An input text-box (maybe larger/resizeable), where the user places a comma-separated list of S2 values to go through. Default might be from 150, in steps of 5, up to 300, and then in steps of 20 to 500, and then in steps of 1000 up to 1000? (*Question: we could also enable Matlab-like notation with “:”, but not sure if it wouldn’t complicate things*)
  + **‘Rate-dependence and alternans:** This simulates the model at a range of frequencies. Again, this will have a distinct section C.
    - **Basic cycle lengths (comma-separated, ms):** A list of bcls that are explored within the simulation. Default might be 250,260,270,…,350, 375, 400, 500, 600,…1000.
  + **‘Delayed afterdepolarizations’:** This runs a train of stimuli, followed by a period of quiescence. *The user will typically want to set high extracellular Ca and strong beta-adrenergic stim, but I probably wouldn’t change this by default…*For section C, this will have the same controls as regular stimulation I think.
    - **DAD type:** Here the user will choose between ‘default’ and ‘stochastic’ DADs. This can be radio buttons or drop-down menu.
    - **Pre-pacing basic cycle length:** A labelled textbox for prepacing bcl
    - **Number of pre-pacing beats:** How many pre-pacing beats are applied
    - **Post-pacing quiescence duration (ms):** After the pre-pacing, this long quiescence should be applied, during which DADs are observed.
* **RUN:** A button (ideally large-ish) in top right that starts the simulation.

# B: Model selection and parameters

* Top row of control elements (left=dropdown, middle and right would be buttons):
  + **Model**: drop-down with endo/epi/mid cell choice. Default is endo.
  + **Save model:** A button that saves the currently selected Myokit model to the user’s hard drive.
  + **Load model:** A button that loads user-provided Myokit model (*we need to make it clear it has to have the same variables/structure as the new model; this will serve for users to make custom adaptations that are not available through sliders etc. in the GUI. That said, we may need to think if this is not hackable e.g. via code injection… If this is a concern, it could be only available in the offline app*).
* Central matrix of controls that parametrize the model:
  + For each variable we want to have controllable, I’d say it would work to have a label, followed by slider (going probably from 0.1 to 3), followed by textbox which shows the value, and can be also edited to input the multiplier manually. For this, we’ll want to have controls of all the major currents, if not all currents... From the top of my head, we’ll definitely want INa, INaL, ICaL, Itof, Itos, INaCa, INaK, IKr, IKs, IK1, Jrel, Jup.
  + Underneath would be labelled textboxes for Cao, Nao, Ko, and Clo.
  + Underneath would be a switch that would control whether beta-adrenergic signalling is on, and if it is, a textbox next to it would be used to input the ISO concentration (probably in uM). *Our Matlab code will have a signalling cAMP pathway (not there yet), which will be switchable on/off – if it’s off, it will save quite a lot of runtime; I don’t know if such a sort of switch can be done in Myokit, but probably yes?*
* *Optional: I’m thinking there could be a bottom row of controls, where one could save the existing starting state and upload a new one, but not really sure whether that’s important enough to implement it?*

# C: Plotting selection

* For regular pacing, EADs, and DADs, the user should be able to select which variables to plot – all ionic currents, and intracellular ionic concentrations (including Ca in the SR). I think checkboxes are probably the best way forward, with membrane potential and Ca being toggled on by default?

For S1S2 and alternans, there should be only the choice of APD90 and CaT amplitude (again, checkboxes)  
D: Plotting window

This is the main plotting window, where simulation-based variables the user selected in section C can be visualized. I think we will want to visualize them one by one, rather than multiple in the same plot. So it could be either that multiple tabs are opened after a simulation finishes, and the user switches between tabs, or there could be a drop-down menu.

For regular pacing and EADs, the user-selected number of last beats is shown (for all the variables toggled)

For DADs, I’d say it’s probably worth plotting last 2-3 beats followed by the full period of quiescence.

For S1S2, the restitution (of either APD or Ca transient amplitude) is shown.

For alternans, a bifurcation of APD90 or Ca transient amplitude is shown (we can plot min and max value in the last 2 beats – if the model is stable, it will overlap, but with alternans, it will show two distinct values).

# E: Output saving

* **Save outputs**: A button which will save simulation outputs as a csv.
* The button above would be accompanied by a drop-down **menu** with the following entries:
  + ‘save toggled’ – saves the outputs that were toggled to be visualized, along with the time vector (in a csv spreadsheet)
  + ‘save all’ – saves all toggleable outputs, as well as state variables (using their myokit names)
* Optional: We could also have something like save/load workplace, where the whole setup including parameter values, selected visualizations etc. could be stored and recalled, but am not sure what’s the ratio of how difficult versus useful this is :)

For alternans – APD or CaT alternans can be ticked