**Scientific overview:**

1. **Is any part of the scientific background unclear or confusing? If so, what additional information would be helpful?**
   1. Explaining the basics of RNA and RNA localization and its importance to human health is good for a general audience.
   2. The dog collar analogy for fluorescent RNA tools is useful for understanding.
   3. Make sure you don’t use terminology like ‘expression gradients’ with a general audience.
   4. Would have been nice to hear more about controls for the imaging.
2. **What part of the project do you find interesting?**
   1. Riboglow is neat.
   2. Using tools to understand stress granule assembly/disassembly is neat.

**Architecture:**

1. **What components of the groups’ proposed architecture do you think is a good design?**
   1. Divided function library based on their use is a good idea.
   2. Having clearly defined inputs and outputs will make this project easier to design.
2. **What, if any, are some limitations of the current proposed architecture that you see?**
   1. Juggling all of the input data will be difficult for sure, but it seems that you have an idea of how to handle this.
   2. Having a big overview schematic for how your code will run might be helpful during the coding process.
   3. The specifics for what each function will do needs to be figured out before you get too far into actual coding.
   4. Figure out all your dependencies (for your plug-ins) before you get too far.
3. **What components of the architecture do you think might be missing?**
   1. Really figure out what TYPE of data your outputs will be and make sure that those will easily flow into the expected inputs of your next function.

**Technical Implementation:**

1. **Do the proposed data types seem suitable for the proposed software design? If not, what could the group improve?**
   1. Yep.
2. **Do you anticipate any computational bottlenecks not described by the group?**
   1. Image processing (segmentation) and creating the masks will likely run really slowly.
3. **Does the delineation of the code development between developers make sense or do you anticipate any code conflicts when merging the code? Does an alternative division of labor seem more suitable?**
   1. Nope. So figure that out before you start writing!