

Lessons Learned from Arbitrum DataGrants Program

By Evan Powell, aka epowell101

Updated March 14, 2024

Introduction:

If you are reading this you likely already know quite a bit about the Arbitrum DataGrants Program. In short - the program is an experimental approach to crowdsource ideas for useful data gathering and analysis via a “permissionless suggestion box”, having submissions to those ideas funded, and reviewed. The experimental program was put together under the auspices of Plurality Labs and their work in establishing the Arbitrum DAO with the input of various OpenData Community members.

The following lessons are by no means comprehensive. They are intended to capture the experience of putting together and operating the DataGrants program so that future program managers and community members can learn from the experience.

This document is being written as the DataGrants Program is kicking off and growing. As such it lacks perspective. I do not know how each aspect of the experiment will work out while I am writing this. Nonetheless, I thought it best to record some details and experiences immediately.

Lessons Learned:

Approval process

- The first lesson learned is simply that the approval process can take some time to go from “yes let’s do it” to actually being approved as a program with initial payments delivered. In my case it took from initial discussions in September 2023 until the end of January 2024.

Data prep vs. data analysis issues and bounties

- Within our scope of looking at Grants and Networks of influence - there are two primary sorts of work: gathering data including datasets and pipelines that others can use on the one hand and on the other hand actually using this data for analysis.
- With that distinction in mind, it likely makes sense to have two sprints, one for data prep including pipelines and the second for analysis
- We did make this split and overall it seemed to work out well; however it would likely be better to make this more clear up front

Dework Settings

- It is useful to set up one or more example issues and to experiment with the Dework settings to see how these settings actually operate.

- In our case, we are setting up the Dework to be as permissionless as possible when it comes to the creation of the issues. This just means the space itself is public and issues appear automatically when they are created on the Github. This should work by default with a Github integration into Dework per the Dework docs. You can choose to have only issues with tags appear in the Dework if you prefer.
- Open to Submission: All issues should be converted to Open to Submission which is the setting within Dework indicating that a bounty like program will be run, with one or more winners identities. Unfortunately Dework requires that you set the number of winners ahead of time. This is unfortunate because we would prefer the council to be able to decide the distribution of the bounty in response to the quality and mix of the responses to the bounties.

Dealing with Duplicate and Overlapping Issues:

- We had a number of duplicate and overlapping issues opened. This of course made the job of the Council to attach bounties to these rather difficult.
- The approach we took was:
 - Create tags for a) data gathering and b) data analysis in Github
 - Reconfigured our Snapshot to only import those issues that have those tags
 - This way we show only those issues that have bounties attached to them
 - Additionally, in a few cases, we needed to write a meta issue, meaning one that perhaps combines the input of two or more issues.

Snapshot settings

- Adding Hats to enable Snapshot is simple. Under settings you go to Strategies. You will see there is a place to paste in the details of the Hats Council you would have established.
- Voting - you'll want to configure Snapshot to use Approval voting.
- Note that in order to set up a new Snapshot, you need an ENS record for that account. In this case I created one for the OpenData Community which, in hindsight, we should already have had. You'll also want to set everything up on Arbitrum of course.

Hats

- With some help from the Hats team, I created a very simple structure for our Hats, which you can see here: <https://app.hatsprotocol.xyz/trees/42161/7?hatId=7.1.1>
- Hats is of course highly extensible and we anticipate that these councils will themselves be voted on, perhaps via Snapchat, in the future.
- Hats is NOT entirely supported as a means of token gating; we found that it would not work for Dework

Discord

- I'm not sure how crucial it is to configure Discord in the way that we have configured it. That said, we are using the existing ODC discord server. Should this program continue in some form, it would likely be best to use its own Discord or perhaps Telegram.

- We have a private section for the Council to discuss and then I try to be extremely transparent in communicating with everyone via a channel we have created in the ODC Discord for the purposes of the DataGrants program.
- We also have weekly meetings which are a new experiment. The goal is to have an office hours session at that time.
- We were very fortunate to have many community members and council members who provided answers and reassurances to the competitors

Spreadsheets

- Because we were unable to grant access via Hats to Dework, we resorted to the creation of spreadsheets for the use of the council members
- This necessitated quite a bit of cutting and pasting. However, it also is very flexible and allows us to create and publish shared criteria for the judging and so on.
-  Bounties for Permissionless Suggestion Box ODC Data Grants