
To: Patrick Barickman
From: Jon Wilkey
Date: February 29, 2016
RE: Monthly report on integrating equipment-based emission factors into R model

During the month of February, work on the R model has focused on (a) updating the UDAQ copy of the R model (to include the changes made to the R model between delivery in Oct. 2015 and the start of this project), (b) importing data from the OGEI database, and (c) preliminary work on the data analysis and simulation of equipment-based emission factors. Each of these items are discussed briefly below.

Updating the UDAQ copy of the R Model

At the conclusion of the previous R modeling project, a final copy of the model was delivered to UDAQ through Bitbucket (a Git version control repository service), and ownership of the repository was transferred to UDAQ. Once work started on incorporating equipment-based emission factors, a decision had to be made to either (a) update the UDAQ repository (so that it included important changes that had been made to the model during Q4 2015) or (b) create a new repository. The group decided that creating a new repository was the best option. The new repository (https://bitbucket.org/wilkey/ub_o-g_emissions) was created, based on the most recent version of the R model used in my thesis¹.

Unrelated components in the model (jobs, taxes, water usage, etc.) were stripped out, download and installation instructions were updated, and UDAQ staff were granted permissions to read and write to the new repository.

Importing data from the OGEI database

I received sample copies of the OGEI database and inventory workbooks at the beginning of February. Initially, I attempted to use a package in R called “RSQLite” to work with each of the tables in the sample OGEI.db file directly. However due to an unknown issue with either RSQLite or the structure of the tables in OGEI.db, I was unable to read from the tables directly without creating serious data errors. For example, when reading from the “apis” table in OGEI.db, R incorrectly interpreted the API numbers in the “api” column so that instead of reading “43-047-52863” it would read “9767844”. Some of the errors may stem from mixed types of variables being located in the same column (e.g. the “api” column is listed as being an integer in the database but includes character hyphens in some of the API number entries). With additional development time, I’m certain that those bugs could be ironed out and connecting directly to the database file would work. However I found that it was faster (for now) to simply switch to importing CSV copies of each table (exported from the OGEI.db using an SQLite browser program). This approach has the downside of requiring an additional user step (i.e. generating the *.csv files), but the result is error-free.

Preliminary data analysis and simulation work

With the data from the OGEI.db file in hand, the majority of my time this month was spent on determining the best method for including equipment-based emission factors in the model’s Monte-Carlo simulation. Originally, I had intended on using the emissions calculated by the formulas in the

¹ Wilkey, J. 2016. “Economic and Environmental Impacts of Oil and Gas Development on the Uinta Basin.” University of Utah.

workbook and database to create relationships between emission sources their total emissions. For example, emissions from well completions (E_{WC} in ton/yr), are calculated in the workbook based on how much diesel fuel is consumed during completion:

$$E_{WC} = \frac{D \cdot EF \cdot (1 - R)}{2000}$$

where D is annual diesel usage (gal/yr), EF is an emissions factor constant (lb/gal), and R is the control percentage (from flaring). The user inputs in the workbook calculation are D and R . Following my original plan, well completion emissions would have been analyzed and simulated by:

1. Creating a cumulative distribution function (CDF) for E_{WC} (based on the results calculated in the workbook/database table).
2. Randomly drawing from that CDF to determine well completion emissions for each well that was completed during the Monte-Carlo simulation.

This effectively folds the two user inputs (D and R) into a single CDF, leveraging the work that has already been put into the workbook and creating a simpler calculation procedure in the R model. However it also eliminates the detailed information that has been collected in the database about both D and R , which in turn reduces the number of ways in which the model's inputs and options can be manipulated and/or investigated by the user. Additionally, there are some equipment types (such as RICE and turbines) for which it would be inappropriate to assume a continuous range of emissions.

Given those drawbacks, I decided to use the same emissions calculation procedure laid out in the workbook in the R model. Under this approach, the algorithm for analyzing the data and simulating emissions for each equipment type is to:

1. Create a CDF for each user input in the workbook for that piece of equipment
2. Randomly draw from each of those CDFs for each well in the Monte-Carlo simulation
3. Calculate the resulting emissions using the equations provided by the workbook

In addition to the advantage of retaining all of the available input information, this approach also allows for verification of the code written in R (by directly comparing emissions results in R with those from the workbook).

At present, functions have been written in R for the calculation of emissions from each of the equipment types listed in the workbook. In order to use these functions, the data analysis steps for each of the user inputs in the workbook must be written. Work in the month of March is expected to focus primarily on that task.