**Remote sensing the gas transfer velocity using SWOT observations**

**Background**

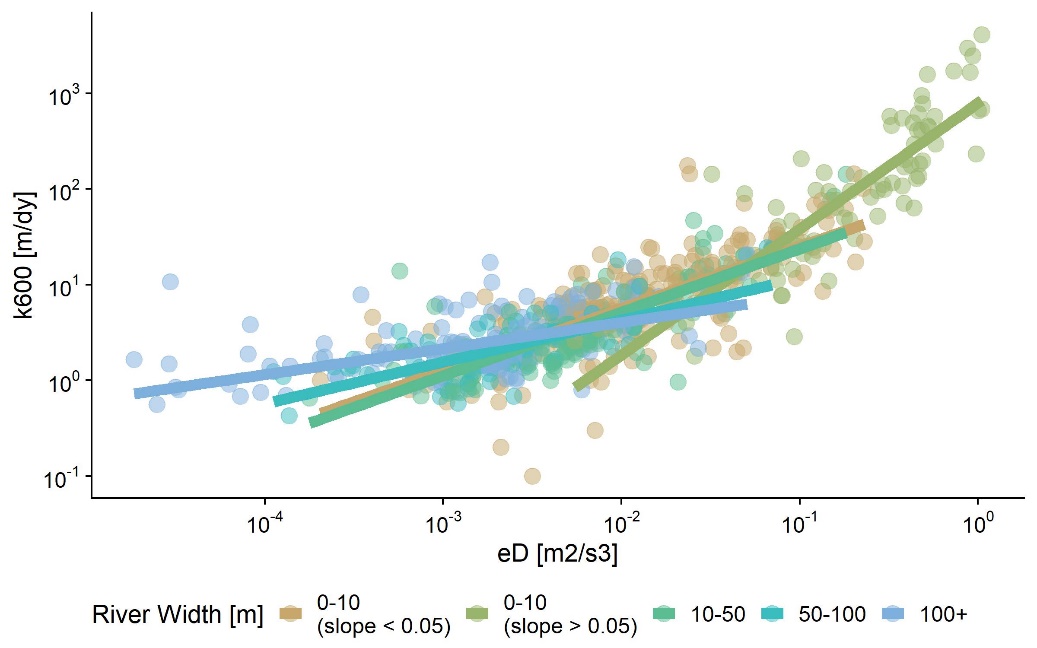
1. CO2 emissions from rivers are significant (duh)
2. Estimating this flux is done by estimating in-stream CO2 and the gas transfer velocity, or ‘evasion rate’ or k600 of the river channel
3. K600 is incredibly cumbersome to directly measure, and impractical when performing upscaling to entire river networks
4. Previous work has stressed that this is largely a f(turbulence) driven by river hydraulics/geomorphology
   1. Empirical scaling functions have been developed to use hydraulics to model k600 for entire river systems but are prone to significant uncertainties (Raymond et al. 2012)
   2. They are also fundamentally dependent on modeled or measured flow velocities which themselves are prone to significant uncertainties (or poorly informed hydraulic geometry models…)
   3. What is needed is a simple way to directly observe the gas transfer velocity at scale and high temporal resolution without needing explicit flow velocity estimates. We argue that the upcoming NASA SWOT mission provides exactly the data required to make this a reality and remotely sense k600 at unprecedented temporal resolution

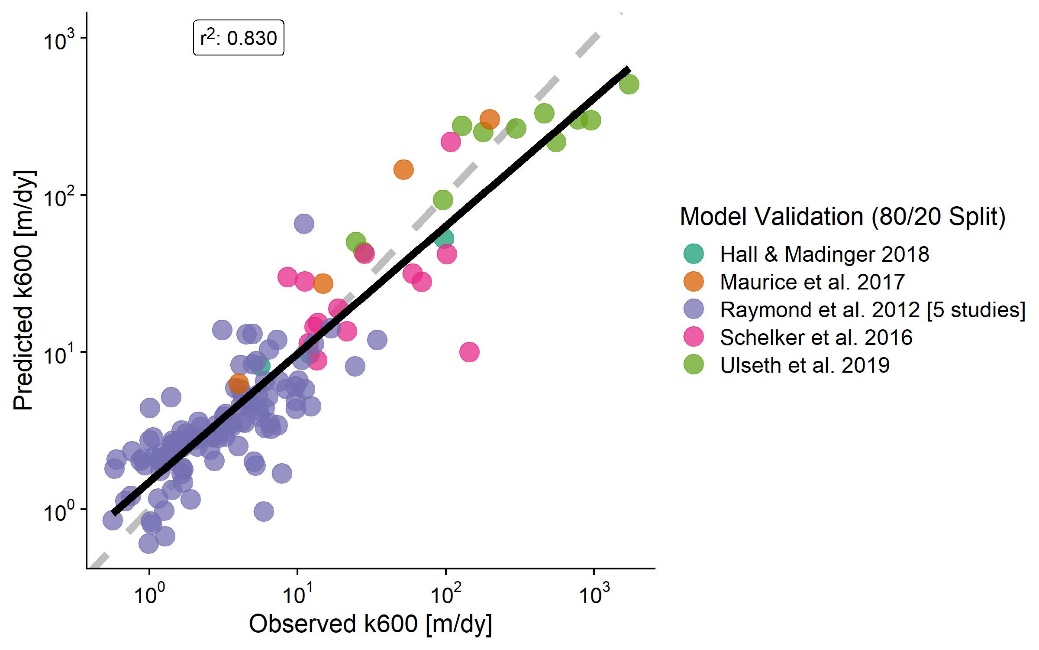
**Research Goals**

1. Formulate a new empirical scaling model to predict k600 using only RS-able geomorphology
2. Build algorithm to run the new empirical model using SWOT observations
3. **?????????**
   1. **Biogeochemists might be interested in a Merritt style test of this algorithm at field-scale (to circumvent velocity measurements…)**
   2. **Pepsi2-style comparisons across temporal sampling?**

**Goal 1**

* Raymond 2012 show that k600 can be scaled using hydraulics b/c k600 generally correlates with the turbulent dissipation rate (eD gSV)
* Ulseth et al 2019 show two distinct scaling regimes for k600 with eD using an expanded dataset including steeper streams than those in Raymond 2012. We further add to this dataset.
* Here, we take the Ulseth et al. 2019 dataset and develop a k600~eD scaling function that accounts for this regime shift but is entirely remotely sensible (i.e. not velocity dependent)
  + where a and b are differentially determined by river size. Thus, from a river’s width and slope alone we can use the most reasonable scaling function.



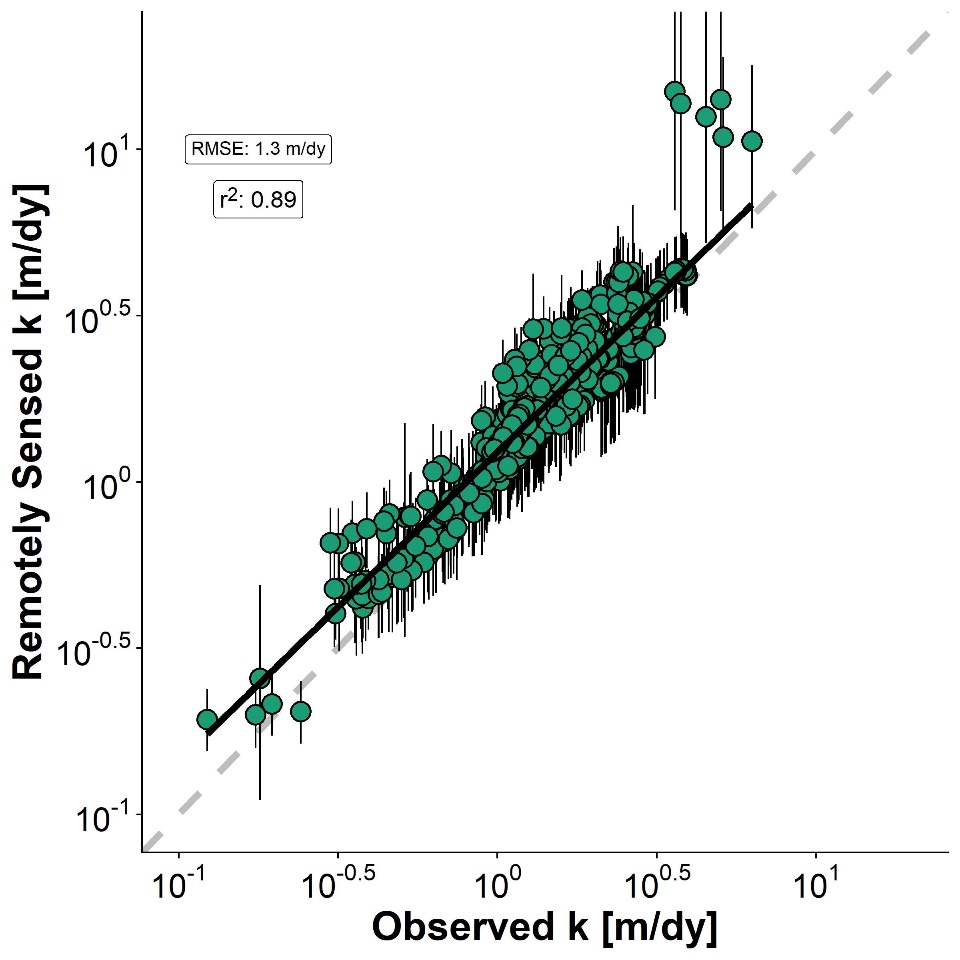


* Monte Carlo simulations to quantify uncertainty from Manning’s equation & k600~eD equation(s)
  + 5,000 random sets of river hydraulics. For each one I ran a 10,000 run Monte Carlo simulation (sampling from parameters), obtaining a histogram of 5,000 Monte Carlo uncertainties

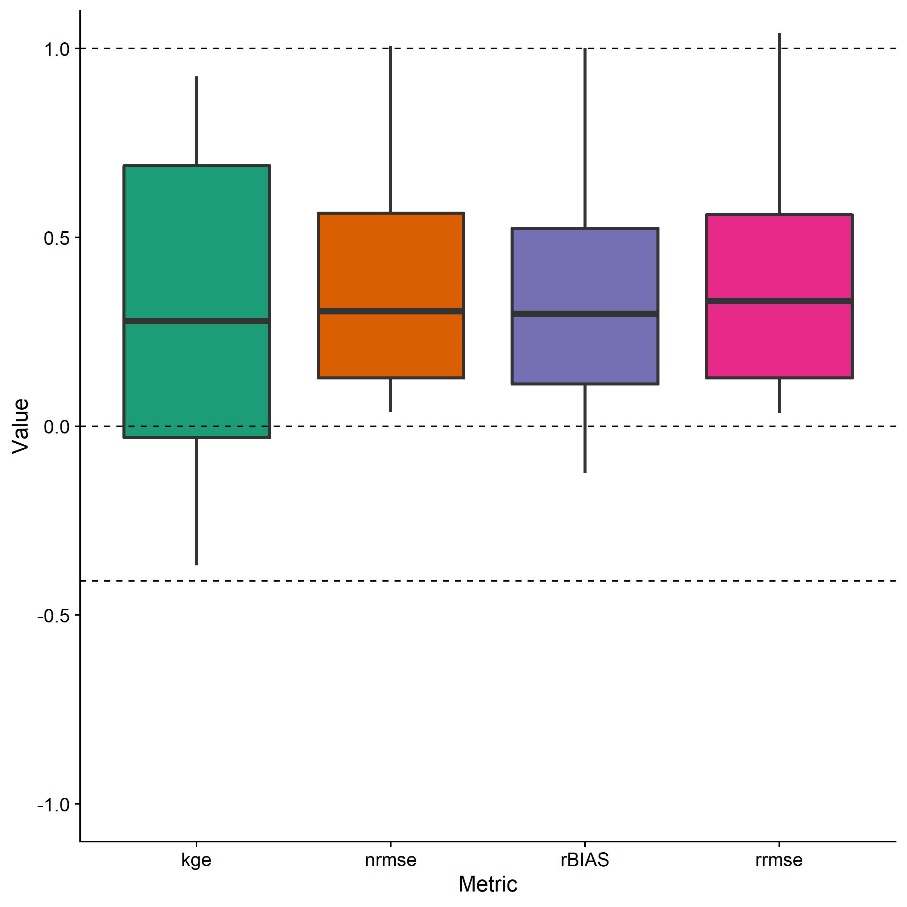
Work in progress

**Goal 2**

* We propose the BIGER (Bayesian Inference/Inversion of Gas Evasion Rate) algorithm to remotely sense k600 from river width/height alone.
* We implement the k600~eD model from Q1 as a Bayesian likelihood function following methods developed for ungauged RSQ (Hagemann et al. 2017)
  + - Model parameters in orange (need priors), data in green
    - a and b are constants, from k600~eD model
    - priors on k600, n, A0
      * geoBAM 🡪 Brinkerhoff etal 2020
      * k600 priors obtained similarly using Ulseth etal 2019
* For the sake of validating the algorithm, we assume that the scaling model is ‘truth’ and see if we can infer these values using only SWOT observables (river width and height)
  + Therefore, in the validation setup, posterior uncertainty is only Manning’s uncertainty and not k600 model uncertainty
  + Validate on 10-day pepsi 2 data. Plotted are means and 95% CIs of the posterior k600 values versus the ‘observed’ k600 calculated from observed velocity and slope for all timesteps across all 32 rivers



* Boxplots of performance metrics for the 32 rivers.
  + There’s a good reason I omit NSE and that -0.41 is noted (Knoben et al. 2019)



**Interesting (though less relevant for this paper)**

-Ulseth et al (2019) fit a piecewise regression with 1 breakpoint (eD of ~0.02)

-The actual optimal number of breakpoints (per r2 ) is 2

The complete breakdown of this scaling relationship below the 1st breakpoint is super clear

