ezmmek tutorial

1 Introduction

Extracellular enzyme assays are a widely-used method to probe the interactions between microbes and complex organic matter. While enzyme assays are popular techniques, there remains a need to standardize the exact methodologies used by practitioners. Here we describe *ezmmek* (Easy Michaelis-Menten Enzyme Kinetics), a package designed to compare common environmental enzyme assay protocols. *ezmmek* is capable of calibrating, calculating, and plotting enzyme activities as they relate to the degradation of synthetic substrates. At present, *ezmmek* contains functions to compare two enzyme assay protocols. The first, as outlined by German *et al.* (2011), accounts for quenching by considering the contribution of each individual component, such as enzyme substrate, buffer standard and homogenate. The second protocol, used by Steen and Arnosti (2011), assumes that the interaction between fluorophore/chromophore standard and homogenate will approximately equal the interaction between that standard's corresponding substrate and homogenate, and result in the same bulk quenching. *ezmmek* contains datasets collected from the same samples via each protocol.

1.1 "German" protocol equations:

$$(1)\ Activity\ (nmol\ g^{-1}\ h^{-1}) = \frac{Net\ Fluorescence * Buffer\ Volume\ (mL)}{Emission\ Coefficient * Homogenate\ Volume\ (mL) * Time\ (h) * Soil\ Mass\ (g)}$$

*For water samples, 'Soil Mass' and 'Homogenate Volume' should both be set to '1'.

$$(2)\ Net\ Fluorescence = \frac{Assay - Homogenate\ Control}{Quench\ Coefficient} - Substrate\ Control$$

$$(3)\ Emission\ Coefficient\ (fluorescence\ nmol^{-1}) = \frac{Standard\ Curve\ Slope\ (in\ presence\ of\ homogenate)\ \left[\frac{Fluorescence}{\frac{nmol}{ml}}\right]}{Assay\ Volume\ (mL)}$$

$$(4)\ Quench\ Coefficient = \frac{Slope\ of\ Standard\ Curve\ (in\ presence\ of\ homogenate)}{Slope\ of\ Standard\ Curve\ (in\ presence\ of\ buffer)}$$

1.2 "Steen" protocol equations:

$$(5)\ Activity\ (nmol\ g^{-1}\ h^{-1}) = \frac{m[Concentration\ of\ Fluorophore\ (nmol) \sim Time\ (h)]}{Soil\ Mass\ (g)}$$

*For water samples, 'Soil Mass' should be set to '1'.

$$(6)\ Concentration\ of\ Fluorophore\ (nmol) = \frac{Assay\ Fluorescence - Kill\ Control - Intercept\ of\ Standard\ Curve\ (in\ presence\ of\ homogenate)}{Slope\ of\ Standard\ Curve\ (in\ presence\ of\ homogenate)}$$

2 Installation

The latest versions of *ezmmek* are available through CRAN and GitHub.

2.1 CRAN

```
install.packages("ezmmek")
```

2.2 GitHub

2.2.1 Install and load *devtools*

```
install.packages("devtools")
library(devtools)
```

2.2.2 Install *ezmmek*

```
install_github("ccook/ezmmek")
```

2.2.3 Required packages

```
assertable
dplyr
magrittr
nls2
purrr
rlang
tidyr
```

3 How to use

3.1 Load *ezmmek*

```
library(ezmmek)
```

3.2 Example datasets

The following datasets stem from samples collected at a pond in Victor Ashe Park, 4901 Bradshaw Road, Knoxville, TN 37912. Analyses were performed using each enzyme assay protocol. The standard curve data file contains data for both protocols. The raw activity data files are split by protocol. In these example datasets, all column names but 'site.name' and 'std.type' are required for *ezmmek* to function properly. While the exact names of 'site.name' and 'std.type' are not required, the user must input at least one descriptor column that is present in both the standard curve dataset and the raw activity dataset for any functions that rely on both datasets.

3.2.1 Standard curve data

```
std_data <- read.csv("data/victor_ashe_std_03172020.csv")
tibble::glimpse(std_data)
#> Observations: 10
#> Variables: 6
#> $ site_name          <fct> victor_ashe, victor_ashe, victor_ashe, victor_ashe…
#> $ std_type           <fct> amc, amc, amc, amc, amc, amc, amc, amc, amc, amc
#> $ std_conc           <int> 0, 0, 1, 1, 2, 2, 3, 3, 4, 4
#> $ homo_signal        <dbl> 4996.60, 4859.96, 229221.03, 222864.45, 429533.03,…
#> $ buffer_signal      <dbl> 413.67, 413.57, 401310.03, 387823.40, 794022.87, 8…
#> $ homo_buffer_signal <dbl> 4996.60, 4859.96, 229221.03, 222864.45, 429533.03,…
```

```
site_name: Name of sampling site, Victor Ashe Park, Knoxville, TN
std_type: Type of standard, 7-Amino-4-methylcoumarin (AMC)
std_conc: Standard concentration, micromolar
homo_signal: Signal from standard in homogenate, fluorescence units
buffer_signal: Signal from standard in buffer, fluorescence units
```

3.2.2 Raw activity data, "german"

```
raw_activity_data_german <- read.csv("data/victor_ashe_sat_german_03172020.csv")
tibble::glimpse(raw_activity_data_german)
#> Observations: 13
#> Variables: 13
#> $ site_name         <fct> victor_ashe, victor_ashe, victor_ashe, victor_ashe,…
#> $ std_type          <fct> amc, amc, amc, amc, amc, amc, amc, amc, amc, amc, a…
#> $ substrate_type    <fct> l-leucine-amc, l-leucine-amc, l-leucine-amc, l-leuc…
#> $ time              <int> 240, 240, 240, 240, 240, 240, 240, 240, 240, 240, 2…
#> $ substrate_conc    <int> 0, 0, 0, 50, 50, 50, 100, 100, 100, 200, 200, 200, …
#> $ signal            <dbl> 4851.65, 5163.49, 5033.46, 295234.09, 162860.89, 15…
#> $ replicate         <int> 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1
#> $ buffer_vol        <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
#> $ homo_vol          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
#> $ soil_mass         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
#> $ assay_vol         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
#> $ homo_control      <dbl> 4982.86, 4977.73, 4981.99, 4982.86, 4977.73, 4981.9…
#> $ substrate_control <dbl> 2137.63, 439.11, 846.38, 25705.65, 25577.37, 27237.…
```

```
site_name: Name of sampling site, Victor Ashe Park, Knoxville, TN
std_type: Type of standard
time: Timepoints at which signal was collected, minutes
signal: Signal from assay, fluorescence units
substrate_conc: Substrate concentration, micromolar
replicate: Sample replicate labeled numerically
homo_vol: Homogenate volume, left as '1' for water samples
soil_mass: Soil mass, left as '1' for water samples
assay_vol: Assay volume, milliliters
homo_control: Homogenate control, fluorescence units
substrate_control: Substrate control, fluorescence units
```

3.2.3 Raw activity data, “steen”

```
raw_activity_data_steen <- read.csv("data/victor_ashe_sat_steen_03172020.csv")
tibble::glimpse(raw_activity_data_steen)
#> Observations: 78
#> Variables: 8
#> $ site_name      <fct> victor_ashe, victor_ashe, victor_ashe, victor_ashe, vi…
#> $ std_type       <fct> amc, amc, amc, amc, amc, amc, amc, amc, amc, amc, amc,…
#> $ substrate_type <fct> l-leucine-amc, l-leucine-amc, l-leucine-amc, l-leucine…
#> $ time           <int> 0, 20, 40, 60, 120, 240, 0, 20, 40, 60, 120, 240, 0, 2…
#> $ substrate_conc <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
#> $ replicate      <int> 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, …
#> $ signal         <dbl> 4905.25, 4830.95, 4933.30, 4849.59, 4918.46, 4851.65, …
#> $ kill_control   <dbl> 8229.68, 8295.77, 8295.47, 8294.22, 8306.66, 8066.26, …
```

```
site_name: Name of sampling site, Victor Ashe Park, Knoxville, TN
std_type: Type of standard
time: Timepoints at which signal was collected, minutes
signal: Signal from assay, fluorescence units
substrate_conc: Substrate concentration, micromolar
replicate: Sample replicated labeled numerically
kill_control: Signal from autoclaved sample, fluorescence units
```

3.3 Visible functions

ezmmek contains several functions that create new data.frame objects. The functions build upon each other. The user can choose how much analyses to perform, from generating standard curve models to calculating km and vmax values. In descending order of parent functions to child functions:

```
new_ezmmek_sat_fit
new_ezmmek_act_calibrate
new_ezmmek_act_group
new_ezmmek_std_group
```

3.3.1 Create data.frame object of class *new_ezmmek_sat_fit*

```
new_obj <- new_ezmmek_sat_fit(std.data.fn,
                              act.data.fn,
                              ...,
                              km = NULL,
                              vmax = NULL,
                              method = NA)
```

```
std.data.fn: Standard curve data file as character string
act.data.fn: Raw activity data file as character string
...: User defined column names to join and group std.data.fn and act.data.fn
km: Starting value to estimate km. Default value is median of 'sub.conc' values
vmax: Starting value to estimate vmax. Default value is max calibrated activity
method: Enzyme assay protocol. Must define method as '"steen"' or '"german"'
```

‘“std.data.fn”’ is the name of the standard curve data file. ‘“act.data.fn”’ is the name of the raw activity data file. ‘…’ are the user-defined column names by which the standard curve and raw activity data files are grouped and joined (*i.e.* descriptor columns like site name and type of fluorophore) to create a new data.frame. The default numeric starting values of ‘km’ and ‘vmax’ are those predicted by *ezmmek*. ‘km’ is calculated as the median substrate concentration used during the experiment. ‘vmax’ is calculated as the maximum calibrated activity value. The user can assign their own ‘km’ and ‘vmax’ starting values if they so choose. ‘method’ must be set equal to ‘“german”’ or ‘“steen”’. The resulting dataframe contains the descriptor columns, standard curve data, raw activity data, calibrated activity data, and Michaelis-Menten fits.

3.3.2 Create data.frame object of class *new_ezmmek_calibrate*

```
new_obj <- new_ezmmek_act_calibrate(std.data.fn,
                                    act.data.fn,
                                    ...,
                                    method = NA,
                                    columns = NULL)
```

```
std.data.fn: Standard curve data file as character string
act.data.fn: Raw activity data file as character string
...: User defined column names to join and group std.data.fn and act.data.fn
method: Enzyme assay protocol. Must define method as '"steen"' or '"german"'
columns: User defined column names to join and group std.data.fn and act.data.fn
```

‘columns’ carries over any ‘…’ arguments from parent functions. If the user does not run this function within a parent function, the ‘columns’ argument can be ignored and left on the ‘NULL’ default. ‘…’ arguments must named if this function is used on its own. The resulting data.frame contains the calibrated activities, but not the Michaelis-Menten fits.

3.3.3 Create data.frame object of class *new_ezmmek_act_group*

```
new_obj <- new_ezmmek_act_group(act.data.fn,
                                ...,
                                method = NA,
                                columns = NULL)
```

```
act.data.fn: Raw activity data file as character string
...: User defined column names to join and group std.data.fn and act.data.fn
method: Enzyme assay protocol. Must define method as '"steen"' or '"german"'
columns: User defined column names to join and group std.data.fn and act.data.fn
```

‘columns’ carries over any ‘…’ arguments from parent functions. If the user does not run this function within a parent function, the ‘columns’ argument can be ignored and left on the ‘NULL’ default. ‘…’ arguments must be named if this function is used on its own. The resulting data.frame contains grouped data of ‘act.data.fn’, as specified by column names.

3.3.4 Create data.frame object of class *new_ezmmek_std_group*

```
new_obj <- new_ezmmek_std_group(std.data.fn,
                                ...,
                                method = NA,
                                columns = NULL)
```

```
std.data.fn: Standard curve data file as character string
...: User defined column names to join and group std.data.fn and act.data.fn
method: Enzyme assay protocol. Must define method as '"steen"' or '"german"'
columns: User defined column names to group std.data.fn
```

'columns' carries over any '…' arguments from parent functions. If the user does not run this function within a parent function, the 'columns' argument can be ignored and left on the 'NULL' default. '…' arguments must be named if this function is used on its own. The resulting data.frame contains grouped 'std.data.fn' data, as specified by column names, and standard curve linear models for each group.

4 Example analyses

4.1 *new_ezmmek_sat_fit*

4.1.1 *new_ezmmek_sat_fit*, "german"

```
new_ezmmek_sat_fit_obj_g <- new_ezmmek_sat_fit("data/victor_ashe_std_03172020.csv",
                                    "data/victor_ashe_sat_german_03172020.csv",
                                    site_name,
                                    std_type,
                                    km = NULL,
                                    vmax = NULL,
                                    method = "german")
#> Joining, by = c("site_name", "std_type")

tibble::glimpse(new_ezmmek_sat_fit_obj_g)
#> Observations: 1
#> Variables: 18
#> $ site_name               <fct> victor_ashe
#> $ std_type                <fct> amc
#> $ substrate_type          <fct> l-leucine-amc
#> $ std_raw_data_g          <list> [<tbl_df[10 x 4]>]
#> $ std_lm_homo_obj         <list> [<-8853.539, 215506.959, 13850.14, 13713.50,…
#> $ std_lm_homo_slope       <dbl> 215507
#> $ std_lm_homo_intercept   <dbl> -8853.539
#> $ std_lm_buffer_obj       <list> [<-20604.44, 428683.42, 21018.107, 21018.007…
#> $ std_lm_buffer_slope     <dbl> 428683.4
#> $ std_lm_buffer_intercept <dbl> -20604.44
#> $ quench_coef             <dbl> 0.5027182
#> $ emission_coef           <dbl> 215507
#> $ act_calibrated_data_g   <list<df[,14]>> 0.000000e+00, 0.000000e+00, 0.00000…
#> $ mm_fit_obj              <list> [<function () , resid, function () , rhs, fu…
#> $ km                      <dbl> 20.52486
#> $ vmax                    <dbl> 0.01072791
#> $ pred_grid               <list> [<data.frame[1000 x 1]>]
#> $ pred_activities         <list> [<data.frame[1000 x 2]>]
```

4.1.2 *new_ezmmek_sat_fit*, "steen"

```
new_ezmmek_sat_fit_obj_s <- new_ezmmek_sat_fit("data/victor_ashe_std_03172020.csv",
                                               "data/victor_ashe_sat_steen_03172020.csv",
                                               site_name,
                                               std_type,
                                               km = NULL,
                                               vmax = NULL,
                                               method = "steen")
#> Joining, by = c("site_name", "std_type")

tibble::glimpse(new_ezmmek_sat_fit_obj_s)
#> Observations: 1
#> Variables: 13
#> $ site_name                    <fct> victor_ashe
#> $ std_type                     <fct> amc
#> $ substrate_type               <fct> l-leucine-amc
#> $ std_raw_data_s               <list> [<tbl_df[10 x 4]>]
#> $ std_lm_homo_buffer_obj       <list> [<-8853.539, 215506.959, 13850.14, 1371…
#> $ std_lm_homo_buffer_slope     <dbl> 215507
#> $ std_lm_homo_buffer_intercept <dbl> -8853.539
#> $ act_calibrated_data_s        <list<df[,9]>> 0.000000e+00, 0.000000e+00, 0.0…
#> $ mm_fit_obj                   <list> [<function () , resid, function () , rh…
#> $ km                           <dbl> 26.28443
#> $ vmax                         <dbl> 0.004830508
#> $ pred_grid                    <list> [<data.frame[1000 x 1]>]
#> $ pred_activities              <list> [<data.frame[1000 x 2]>]
```

4.2 *new_ezmmek_act_calibrate*

4.2.1 *new_ezmmek_act_calibrate*, “german”

```
new_ezmmek_act_calibrate_obj_g <- new_ezmmek_act_calibrate("data/victor_ashe_std_03172020.csv",
                                                           "data/victor_ashe_sat_german_03172020
                                                           site_name,
                                                           std_type,
                                                           method = "german",
                                                           columns = NULL)
#> Joining, by = c("site_name", "std_type")

tibble::glimpse(new_ezmmek_act_calibrate_obj_g)
#> Observations: 1
#> Variables: 13
#> $ site_name               <fct> victor_ashe
#> $ std_type                <fct> amc
#> $ substrate_type          <fct> l-leucine-amc
#> $ std_raw_data_g          <list> [<tbl_df[10 x 4]>]
#> $ std_lm_homo_obj         <list> [<-8853.539, 215506.959, 13850.14, 13713.50,…
#> $ std_lm_homo_slope       <dbl> 215507
#> $ std_lm_homo_intercept   <dbl> -8853.539
#> $ std_lm_buffer_obj       <list> [<-20604.44, 428683.42, 21018.107, 21018.007…
#> $ std_lm_buffer_slope     <dbl> 428683.4
#> $ std_lm_buffer_intercept <dbl> -20604.44
#> $ quench_coef             <dbl> 0.5027182
#> $ emission_coef           <dbl> 215507
#> $ act_calibrated_data_g   <list<df[,14]>> 0.000000e+00, 0.000000e+00, 0.00000…
```

4.2.2 *new_ezmmek_act_calibrate*, “steen”

```
new_ezmmek_act_calibrate_obj_s <- new_ezmmek_act_calibrate("data/victor_ashe_std_03172020.csv",
                                                           "data/victor_ashe_sat_steen_03172020.cs
                                                           site_name,
                                                           std_type,
                                                           method = "steen",
                                                           columns = NULL)
#> Joining, by = c("site_name", "std_type")

tibble::glimpse(new_ezmmek_act_calibrate_obj_s)
#> Observations: 1
#> Variables: 8
#> $ site_name                   <fct> victor_ashe
#> $ std_type                    <fct> amc
#> $ substrate_type              <fct> l-leucine-amc
#> $ std_raw_data_s              <list> [<tbl_df[10 x 4]>]
#> $ std_lm_homo_buffer_obj      <list> [<-8853.539, 215506.959, 13850.14, 1371…
#> $ std_lm_homo_buffer_slope    <dbl> 215507
#> $ std_lm_homo_buffer_intercept <dbl> -8853.539
#> $ act_calibrated_data_s       <list<df[,9]>> 0.000000e+00, 0.000000e+00, 0.0…
```

4.3 *new_ezmmek_act_group*

4.3.1 *new_ezmmek_act_group*, “german”

```
new_ezmmek_act_group_obj_g <- new_ezmmek_act_group("data/victor_ashe_sat_german_03172020.csv",
                                                   site_name,
                                                   std_type,
                                                   method = "german",
                                                   columns = NULL)

tibble::glimpse(new_ezmmek_act_group_obj_g)
#> Observations: 1
#> Variables: 3
#> $ site_name      <fct> victor_ashe
#> $ std_type       <fct> amc
#> $ act_raw_data_g <list> [<tbl_df[13 x 11]>]
```

4.3.2 *new_ezmmek_act_group*, “steen”

```
new_ezmmek_act_group_obj_s <- new_ezmmek_act_group("data/victor_ashe_sat_steen_03172020.csv",
                                                   site_name,
                                                   std_type,
                                                   method = "steen",
                                                   columns = NULL)

tibble::glimpse(new_ezmmek_act_group_obj_s)
#> Observations: 1
#> Variables: 3
#> $ site_name      <fct> victor_ashe
#> $ std_type       <fct> amc
#> $ act_raw_data_s <list> [<tbl_df[78 x 6]>]
```

4.4 *new_ezmmek_std_group*

4.4.1 *new_ezmmek_std_group*, “german”

```
new_ezmmek_std_group_obj_g <- new_ezmmek_std_group("data/victor_ashe_std_03172020.csv",
                                                   site_name,
                                                   std_type,
                                                   method = "german",
                                                   columns = NULL)

tibble::glimpse(new_ezmmek_std_group_obj_g)
#> Observations: 1
#> Variables: 10
#> $ site_name               <fct> victor_ashe
#> $ std_type                <fct> amc
#> $ std_raw_data_g          <list> [<tbl_df[10 x 4]>]
#> $ std_lm_homo_obj         <list> [<-8853.539, 215506.959, 13850.14, 13713.50,…
#> $ std_lm_homo_slope       <dbl> 215507
#> $ std_lm_homo_intercept   <dbl> -8853.539
#> $ std_lm_buffer_obj       <list> [<-20604.44, 428683.42, 21018.107, 21018.007…
#> $ std_lm_buffer_slope     <dbl> 428683.4
#> $ std_lm_buffer_intercept <dbl> -20604.44
#> $ quench_coef             <dbl> 0.5027182
```

4.4.2 *new_ezmmek_std_group*, “steen”

```
new_ezmmek_std_group_obj_s <- new_ezmmek_std_group("data/victor_ashe_std_03172020.csv",
                                                   site_name,
                                                   std_type,
                                                   method = "german",
                                                   columns = NULL)
tibble::glimpse(new_ezmmek_std_group_obj_s)
#> Observations: 1
#> Variables: 10
#> $ site_name               <fct> victor_ashe
#> $ std_type                <fct> amc
#> $ std_raw_data_g          <list> [<tbl_df[10 x 4]>]
#> $ std_lm_homo_obj         <list> [<-8853.539, 215506.959, 13850.14, 13713.50,…
#> $ std_lm_homo_slope       <dbl> 215507
#> $ std_lm_homo_intercept   <dbl> -8853.539
#> $ std_lm_buffer_obj       <list> [<-20604.44, 428683.42, 21018.107, 21018.007…
#> $ std_lm_buffer_slope     <dbl> 428683.4
#> $ std_lm_buffer_intercept <dbl> -20604.44
#> $ quench_coef             <dbl> 0.5027182
```

5 Methods for *new_ezmmek* objects

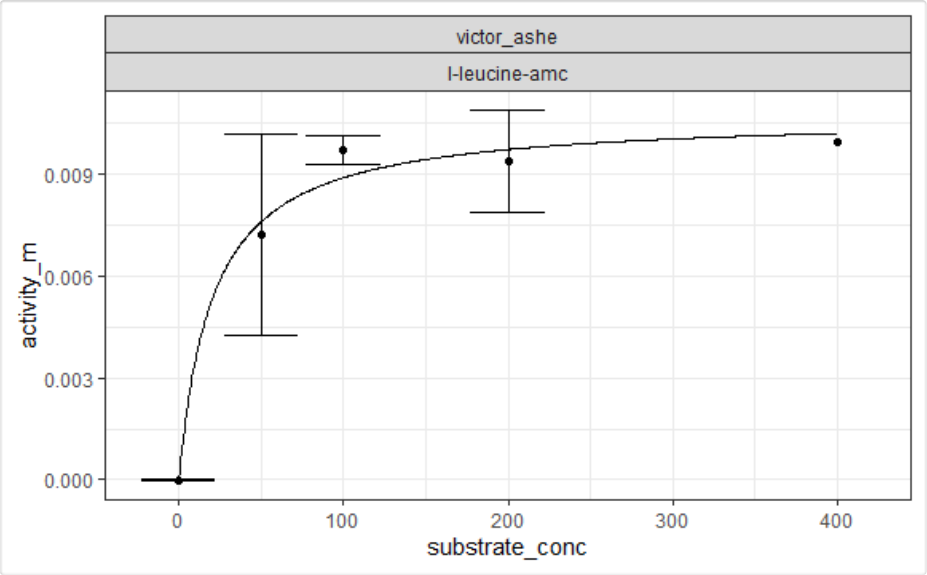
5.1 *plot*

plot methods were created for each *new_ezmmek* object. The user can choose how to facet their plots by adding those columns/variables as arguments. Because each *new_ezmmek* object is built off another, the child functions can be applied to objects created by more parent functions. For example, one can plot a standard curve using *plot.new_ezmmek_std_group(new_ezmmek_sat_fit_obj_g)*.

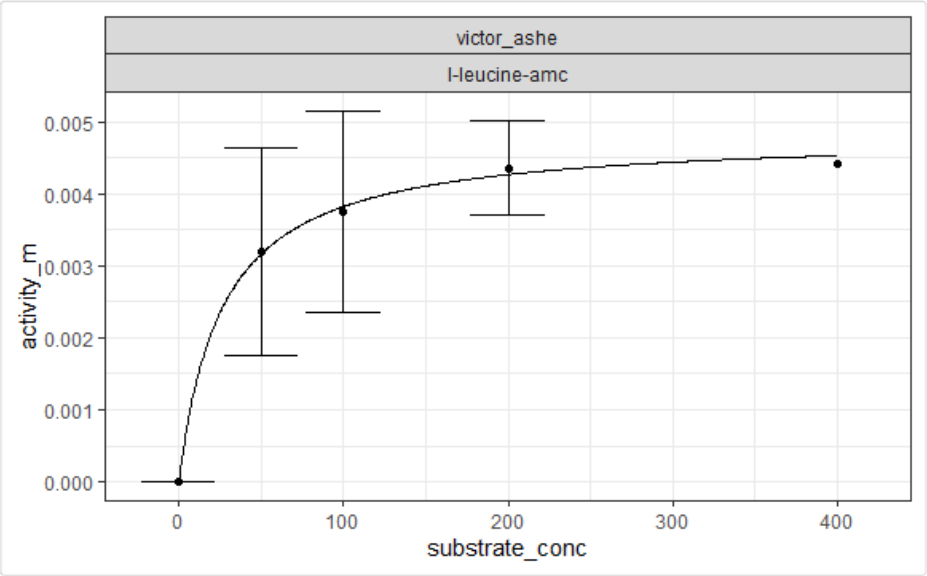
5.1.1 *new_ezmmek_sat_fit*, “german” and “steen”

Plot overlay of saturation curve on activities

```
plot(new_ezmmek_sat_fit_obj_g, site_name, substrate_type)
```



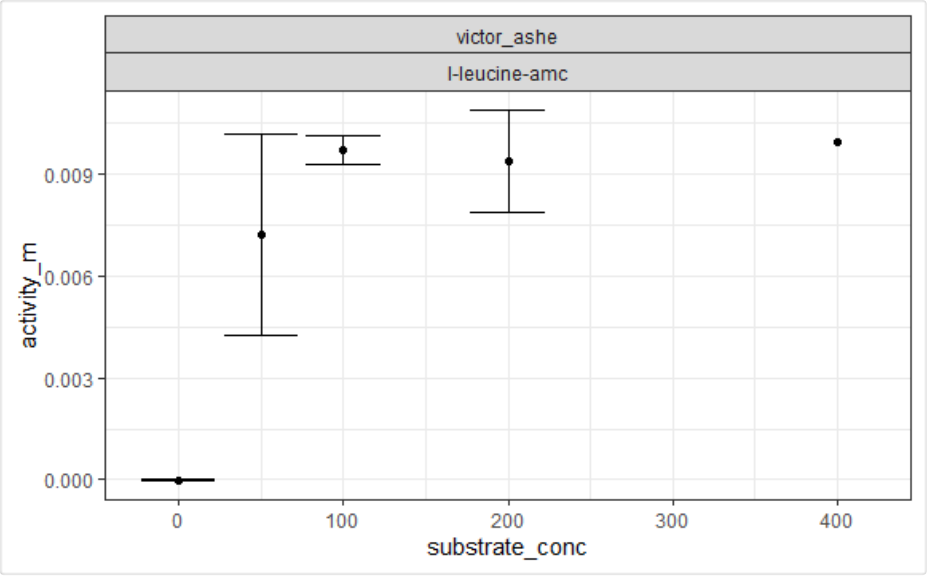

```
plot(new_ezmmek_sat_fit_obj_s, site_name, substrate_type)
```



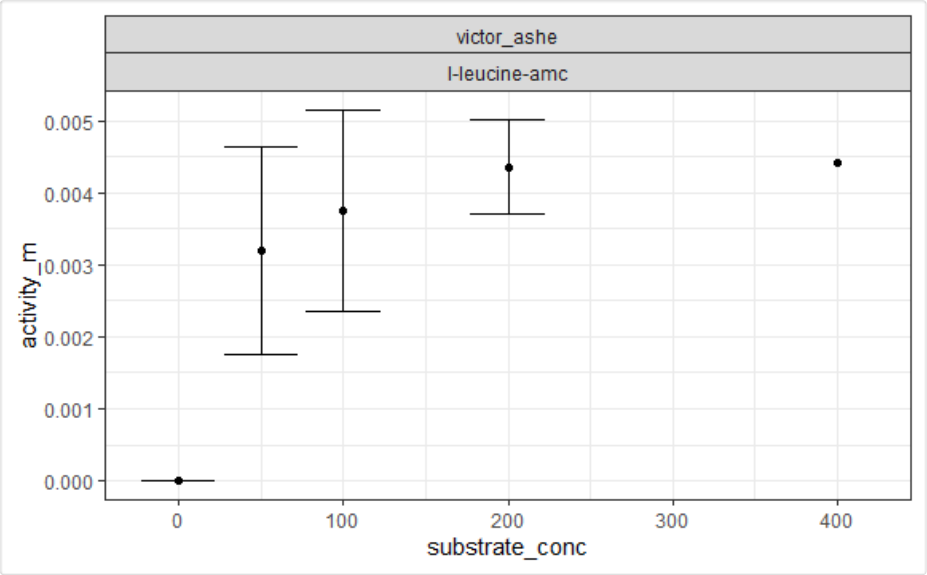
5.1.2 *new_ezmmek_calibrate*, "german" and "steen"

Plot activities and their associated error

```
plot(new_ezmmek_act_calibrate_obj_g, site_name, substrate_type)
```



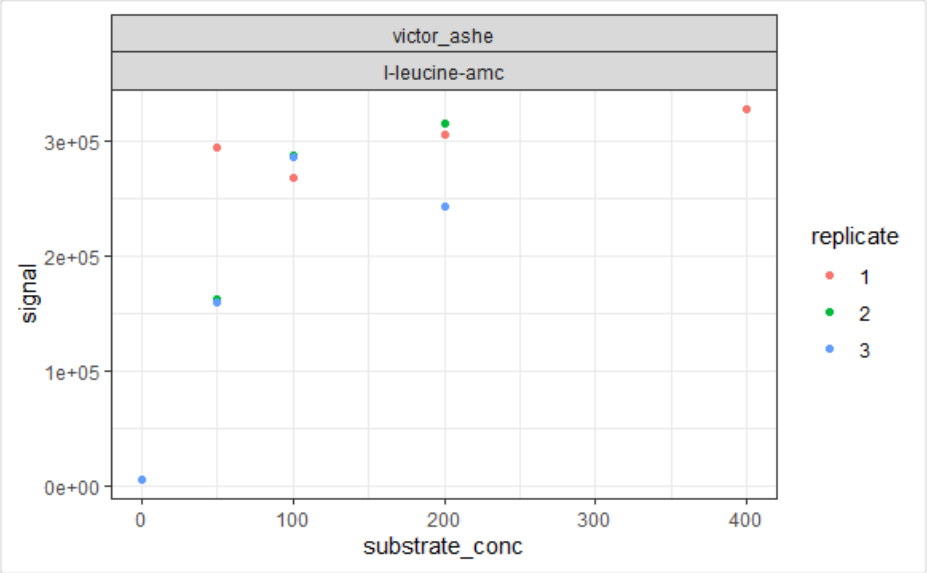
```
plot(new_ezmmek_act_calibrate_obj_s, site_name, substrate_type)
```



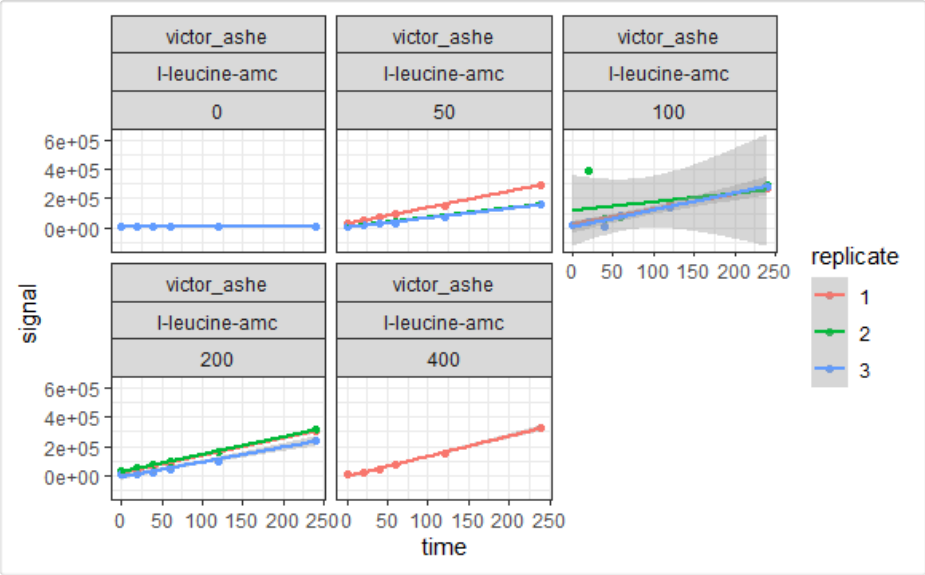
5.1.3 *new_ezmmek_act_group*, "german" and "steen"

Plot raw activity data

```
plot(new_ezmmek_act_group_obj_g, site_name, substrate_type)
```



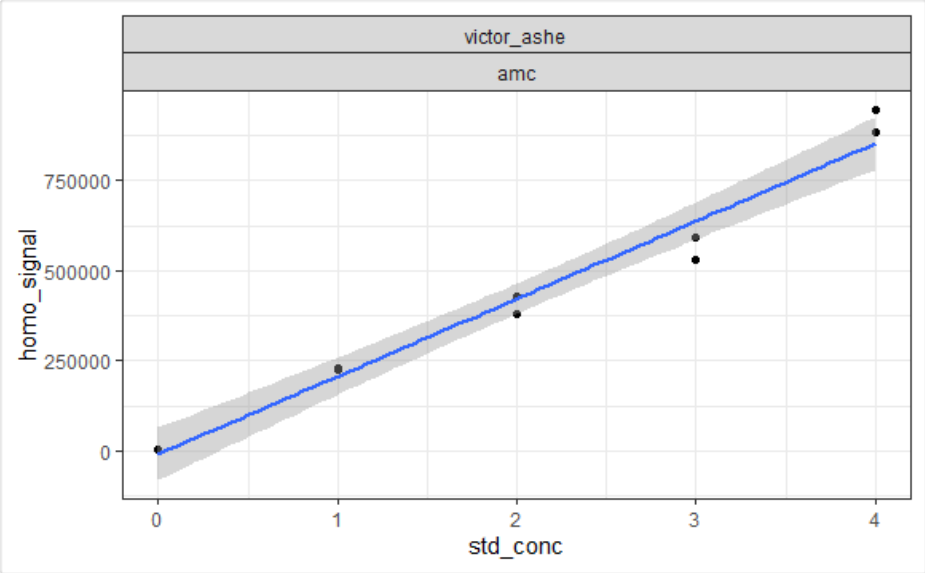

```
plot(new_ezmmek_act_group_obj_s, site_name, substrate_type, substrate_conc)
```

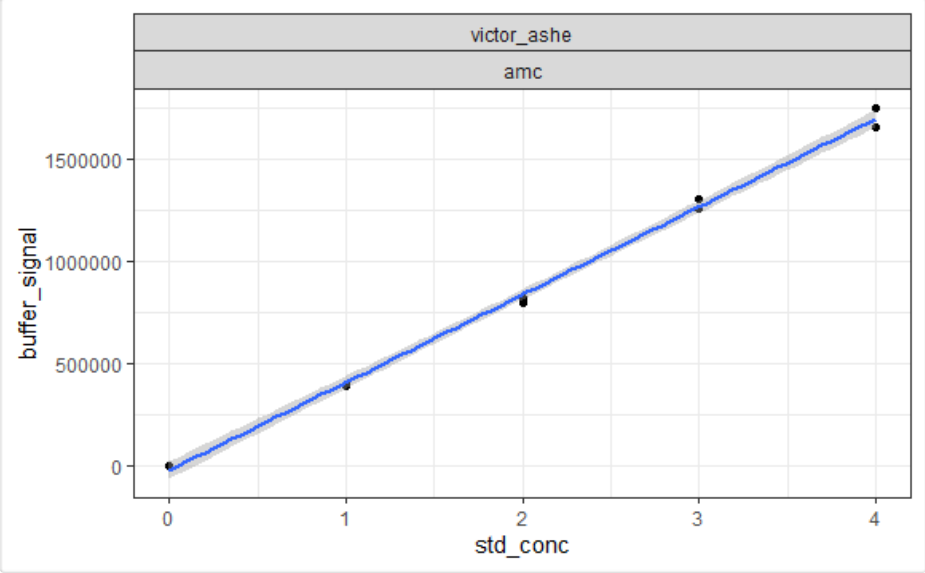


5.1.4 *new_ezmmek_std_group*, "german" and "steen"

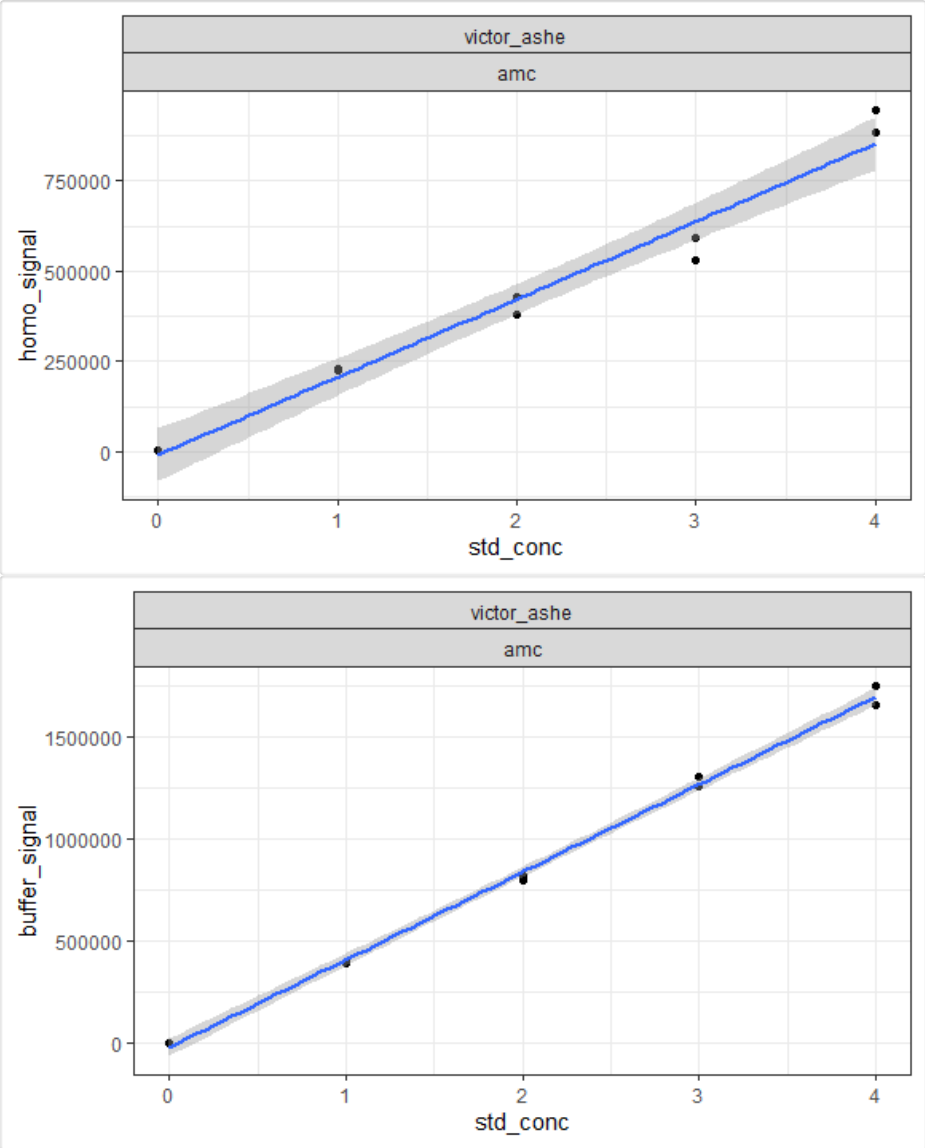
Plot standard curve(s)

```
plot(new_ezmmek_std_group_obj_g, site_name, std_type)
```






```
plot(new_ezmmek_std_group_obj_s, site_name, std_type)
```



5.2 *summary*

5.2.1 *new_ezmmek_sat_fit*, "german" and "steen"

6 References

German *et al.* (2011) doi:10.1016/j.soilbio.2011.03.017
Sinsabaugh *et al.* (2014) doi:10.1007/s10533-014-0030-y
Steen and Arnosti (2011) doi:10.1016/j.marchem.2010.10.006

7 Authors

Christopher L. Cook (ccook62@vols.utk.edu) and Andrew D. Steen

8 License