NCA Tutorial

Yingbo Ma

July 19, 2019

1 Introduction

This is an introduction to `NCA.jl`, a software for noncompartmental analysis (NCA). In this tutorial we will show how to use `NCA.jl` to analysis data.

1.1 Installation

Currently, `NCA.jl` is a submodule in `Pumas.jl`, so you only need to install `Pumas.jl`, and everything will be ready to go.

1.2 Getting Started

To load the package, use

```
using Pumas.NCA
```

First, let's load the example NCA data inside `Pumas.jl`. This data have 24 individuals, and each of them has 16 data points.

```
using Pumas, CSV

file = Pumas.example_nmtran_data("nca_test_data/dapa_IV")
data = CSV.read(file)
```

	ID	TIME	TAD	CObs	AMT_IV	AMT_ORAL	Formulation
	Int64	Float64	Float64	Float64	Float64	Float64	String
1	1	0.0	0.0	157.021	5000.0	0.0	IV
2	1	0.05	0.05	141.892	0.0	0.0	IV
3	1	0.35	0.35	116.228	0.0	0.0	IV
4	1	0.5	0.5	109.353	0.0	0.0	IV
5	1	0.75	0.75	66.4814	0.0	0.0	IV
6	1	1.0	1.0	74.7532	0.0	0.0	IV
7	1	2.0	2.0	39.1933	0.0	0.0	IV
8	1	3.0	3.0	25.4495	0.0	0.0	IV
9	1	4.0	4.0	13.0165	0.0	0.0	IV
10	1	6.0	6.0	3.81448	0.0	0.0	IV
11	1	8.0	8.0	1.47339	0.0	0.0	IV
12	1	10.0	10.0	1.10532	0.0	0.0	IV
13	1	12.0	12.0	0.911367	0.0	0.0	IV
14	1	16.0	16.0	0.830115	0.0	0.0	IV
15	1	20.0	20.0	0.624201	0.0	0.0	IV
16	1	24.0	24.0	0.653632	0.0	0.0	IV
17	2	0.0	0.0	59.7702	5000.0	0.0	IV
18	2	0.05	0.05	66.354	0.0	0.0	IV
19	2	0.35	0.35	55.507	0.0	0.0	IV
20	2	0.5	0.5	59.0243	0.0	0.0	IV
21	2	0.75	0.75	55.8154	0.0	0.0	IV
22	2	1.0	1.0	53.6728	0.0	0.0	IV
23	2	2.0	2.0	38.8955	0.0	0.0	IV
24	2	3.0	3.0	30.9587	0.0	0.0	IV
25	2	4.0	4.0	24.2407	0.0	0.0	IV
26	2	6.0	6.0	15.8675	0.0	0.0	IV
27	2	8.0	8.0	10.663	0.0	0.0	IV
28	2	10.0	10.0	7.32787	0.0	0.0	IV
29	2	12.0	12.0	5.83294	0.0	0.0	IV
30	2	16.0	16.0	3.30032	0.0	0.0	IV
31	2	20.0	20.0	2.32031	0.0	0.0	IV
32	2	24.0	24.0	1.71656	0.0	0.0	IV
33	3	0.0	0.0	165.733	5000.0	0.0	IV
34	3	0.05	0.05	130.022	0.0	0.0	IV
35	3	0.35	0.35	127.35	0.0	0.0	IV
36	3	0.5	0.5	97.7563	0.0	0.0	IV
37	3	0.75	0.75	86.6491	0.0	0.0	IV
38	3	1.0	1.0	81.895	0.0	0.0	IV
39	3	2.0	2.0	35.7601	0.0	0.0	IV
40	3	3.0	3.0	22.2936	0.0	0.0	IV
41	3	4.0	4.0	12.7924	0.0	0.0	IV
42	3	6.0	6.0	6.46943	0.0	0.0	IV
43	3	8.0	8.0	4.9808	0.0	0.0	IV
44	3	10.0	10.0	3.38318	0.0	0.0	IV
45	3	12.0	12.0	3.33256	0.0	0.0	IV
46	3	16.0	16.0	2.69385	0.0	0.0	IV
47	3	20.0	20.0	2.21823	0.0	0.0	IV
48	3	24.0	24.0	2.0378	0.0	0.0	IV
49	4	0.0	0.0	133.911	5000.0	0.0	IV
50	4	0.05	0.05	123.97	0.0	0.0	IV
51	4	0.35	0.35	111.766	0.0	0.0	IV
52	4	0.5	0.5	122.325	0.0	0.0	IV

here is what the dataset looks like

```julia
first(data, 6) # take first 6 rows
```

	ID	TIME	TAD	CObs	AMT_IV	AMT_ORAL	Formulation
	Int64	Float64	Float64	Float64	Float64	Float64	String
1	1	0.0	0.0	157.021	5000.0	0.0	IV
2	1	0.05	0.05	141.892	0.0	0.0	IV
3	1	0.35	0.35	116.228	0.0	0.0	IV
4	1	0.5	0.5	109.353	0.0	0.0	IV
5	1	0.75	0.75	66.4814	0.0	0.0	IV
6	1	1.0	1.0	74.7532	0.0	0.0	IV

2 Efficient Computation of Multiple NCA Diagnostics

2.1 AUC and AUMC

We can compute the area under the curve (AUC) from the first observation time to infinity. Below we are accessing the concentration and corresponding time array for the first individual. By default, the `auc` function computes the AUC from initial time to infinity (AUCinf).

```julia
NCA.auc(data[:CObs][1:16], data[:TIME][1:16])
```

```
263.792662196049
```

```julia
NCA.auc(data[:CObs][1:16], data[:TIME][1:16], method=:linuplogdown)
```

```
257.8792837435675
```

the keyword argument `method` can be `:linear`, `:linuplogdown`, or `:linlog`, and it defaults to `:linear`. This is a simple interface, however it is not efficient if you want to compute many quantities. The recommended way is to create an `NCASubject` or an `NCAPopulation` object first and then call the respective NCA diagnostic on the data object. To parse data to an `NCAPopulation` object one can call the `read_nca` function and assign the corresponding data to column names: `id`, `time`, `conc` (concentration), `amt` (dosage), and `route`. Note that, by default, the lower limit of quantization (LLQ) is 0, and concentrations that are below LLQ (BLQ) are dropped. Also, we can add units by providing `timeu`, `concu`, and `amtu`.

```julia
timeu = u"hr"
concu = u"mg/L"
amtu  = u"mg"
data.id = data.ID
data.time = data.TIME
data.conc = data.CObs
data.amt = data.AMT_IV
data.route = "iv"
pop = read_nca(data, llq=0concu, timeu=timeu, concu=concu, amtu=amtu)
```

```
NCAPopulation (24 subjects):
  ID: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2
0, 21, 22, 23, 24]
    concentration: mg L^-1
    time:          hr
    auc:           mg hr L^-1
    aumc:          mg hr^2 L^-1
```

```
λz:          hr^-1
dose:        mg
```

Here, each element of `pop` has the type `NCASubject`. It is a lazy data structure and actual computations are not performed. When we are instantiating `NCASubject`, it only performs data checking and cleaning. To calculate AUC, one can do:

```
NCA.auc(pop)
```

	id	auc
	Int64	Quantity
1	1	263.793 mg hr L^-1
2	2	323.253 mg hr L^-1
3	3	339.848 mg hr L^-1
4	4	373.361 mg hr L^-1
5	5	132.145 mg hr L^-1
6	6	303.86 mg hr L^-1
7	7	380.275 mg hr L^-1
8	8	279.126 mg hr L^-1
9	9	239.831 mg hr L^-1
10	10	260.862 mg hr L^-1
11	11	146.864 mg hr L^-1
12	12	359.489 mg hr L^-1
13	13	522.905 mg hr L^-1
14	14	262.988 mg hr L^-1
15	15	378.993 mg hr L^-1
16	16	206.926 mg hr L^-1
17	17	341.551 mg hr L^-1
18	18	195.925 mg hr L^-1
19	19	433.443 mg hr L^-1
20	20	214.27 mg hr L^-1
21	21	232.537 mg hr L^-1
22	22	471.515 mg hr L^-1
23	23	292.413 mg hr L^-1
24	24	170.305 mg hr L^-1

`AUClast` is the area under the curve from the first observation to the last observation. To compute `AUClast` on the second individual, one would do:

```
NCA.auc(pop[2], auctype=:last)
```

```
302.24594 mg hr L^-1
```

Or to compute the AUC on every individual, one would do:

```
NCA.auc(pop, auctype=:last)
```

	id	auc
	Int64	Quantity
1	1	246.932 mg hr L^-1
2	2	302.246 mg hr L^-1
3	3	288.58 mg hr L^-1
4	4	333.804 mg hr L^-1
5	5	129.061 mg hr L^-1
6	6	291.951 mg hr L^-1
7	7	333.994 mg hr L^-1
8	8	259.967 mg hr L^-1
9	9	233.643 mg hr L^-1
10	10	242.719 mg hr L^-1
11	11	141.435 mg hr L^-1
12	12	311.005 mg hr L^-1
13	13	427.174 mg hr L^-1
14	14	246.329 mg hr L^-1
15	15	311.131 mg hr L^-1
16	16	196.672 mg hr L^-1
17	17	319.297 mg hr L^-1
18	18	185.399 mg hr L^-1
19	19	403.216 mg hr L^-1
20	20	202.64 mg hr L^-1
21	21	222.77 mg hr L^-1
22	22	364.756 mg hr L^-1
23	23	265.663 mg hr L^-1
24	24	156.806 mg hr L^-1

One can also compute AUC on a certain interval. To compute AUC on the interval $[10, \infty]$ on the first individual

```
NCA.auc(pop[1], interval=(10,Inf).*timeu)
```

```
27.824427196048966 mg hr L^-1
```

Note that we need to apply the time unit to the interval for units compatibility. One can also specify multiple intervals

```
NCA.auc(pop[1], interval=[(10,Inf).*timeu, (10, 15).*timeu])
```

```
2-element Array{Quantity{Float64,M*T*L^-3,Unitful.FreeUnits{(mg, hr, L^-1),
M*T*L^-3,nothing}},1}:
 27.824427196048966 mg hr L^-1
          4.6593795 mg hr L^-1
```

In many cases, the AUC commands may need to extrapolate in order to cover the desired interval. To see the percentage of extrapolation ($\frac{\text{extrapolated AUC}}{\text{Total AUC}} \cdot 100$), you can use the command:

```
NCA.auc_extrap_percent(pop[1])
```

```
6.391564517256502
```

Area under the first moment of the concentration (AUMC) is

$$\int_{t_0}^{t_1} t \cdot \text{concentration}(t) dt.$$

The interface of computing AUMC is exactly the same with AUC, and one needs to change `auc` to `aumc` for calculating AUMC or related quantities. For instance,

```
NCA.aumc_extrap_percent(pop[1])
NCA.aumc(pop[1])
```

```
1411.6198735770822 mg hr^2 L^-1
```

2.2 Terminal Rate Constant (λz)

The negative slope for concentration vs time in log-linear scale is the terminal rate constant, often denoted by λz. To compute λz, one can call

```
NCA.lambdaz(pop[1])
```

```
0.03876710923615265 hr^-1
```

To get the coefficient of determination (r^2), the adjusted coefficient of determination ($adjr^2$), the y-intercept, the first time point used, and the number of points used while computing λz, one can do:

```
NCA.lambdazr2(pop)
NCA.lambdazadjr2(pop)
NCA.lambdazintercept(pop)
NCA.lambdaztimefirst(pop)
NCA.lambdaznpoints(pop)
```

	id	lambdaznpoints
	Int64	Int64
1	1	5
2	2	3
3	3	5
4	4	6
5	5	5
6	6	10
7	7	4
8	8	4
9	9	7
10	10	4
11	11	6
12	12	5
13	13	4
14	14	6
15	15	3
16	16	6
17	17	4
18	18	5
19	19	7
20	20	3
21	21	5
22	22	3
23	23	4
24	24	3

By default, λz calculation checks last 10 or less data points, one can change it by providing the keyword `threshold`, e.g.

```
NCA.lambdaz(pop[1], threshold=3)
```

```
0.029877467931765923 hr^-1
```

One can also specify the exact data points by passing their indices

```
NCA.lambdaz(pop[1], idxs=[10, 15, 16])
```

```
0.10617388957053892 hr^-1
```

You can also pass their time points

```
NCA.lambdaz(pop[1], slopetimes=[1,2,3].*timeu)
```

```
0.5387479621404708 hr^-1
```

2.3 Simple functions

`T_max` is the time point at which the maximum concentration (`C_max`) is observed, and they can be computed by:

```
NCA.tmax(pop[1])
NCA.cmax(pop[1])
NCA.cmax(pop[1], interval=(20, 24).*timeu)
NCA.cmax(pop[1], interval=[(20, 24).*timeu, (10, 15).*timeu])
```

```
2-element Array{Quantity{Float64,M*L^-3,Unitful.FreeUnits{(mg, L^-1),M*L^-3
,nothing}},1}:
 0.653632 mg L^-1
  1.10532 mg L^-1
```

Note that `cmax` returns `C_max` and normalized `C_max` if `dose` is provided. If `dose` is provided in the `NCASubject`, that `dose` will be used by all computations where dose can be used.

`T_last` is the time of the last observed concentration value above the lower limit of quantization (LLQ), and the corresponding concentration value is (`C_last`). They can be computed by the command

```
NCA.tlast(pop[1])
NCA.clast(pop[1])
```

```
0.653632 mg L^-1
```

The half-life can be computed by:

```
NCA.thalf(pop[1])
```

```
1.2865889604594312 hr
```

One may need to interpolate or to extrapolate the concentration-time data. For example, if you wanted to interpolate the concentration at $t = 12$ using linear interpolation, you would do:

```
NCA.interpextrapconc(pop[1], 12timeu, method=:linear)
```

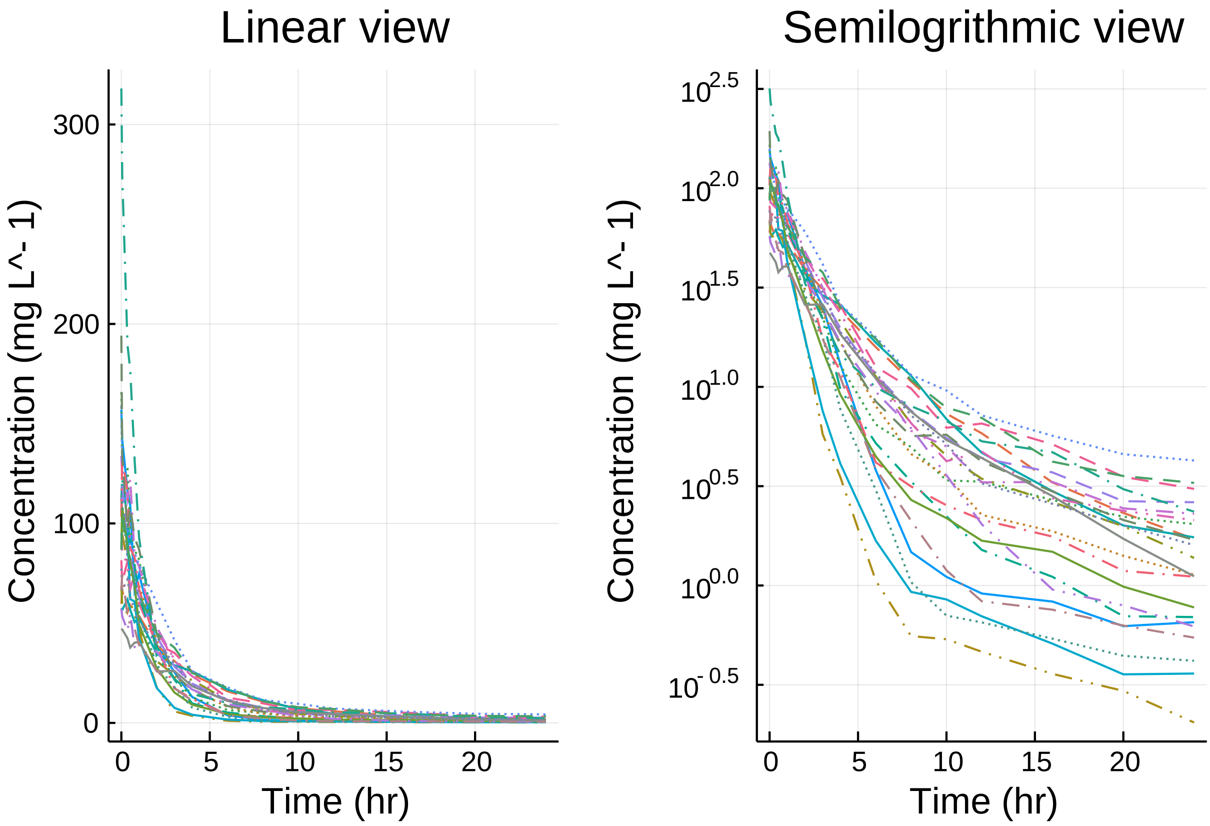
```
0.911367 mg L^-1
```

`method` can be `:linear`, `:linuplogdown`, or `:linlog`.

3 Plots and Summary

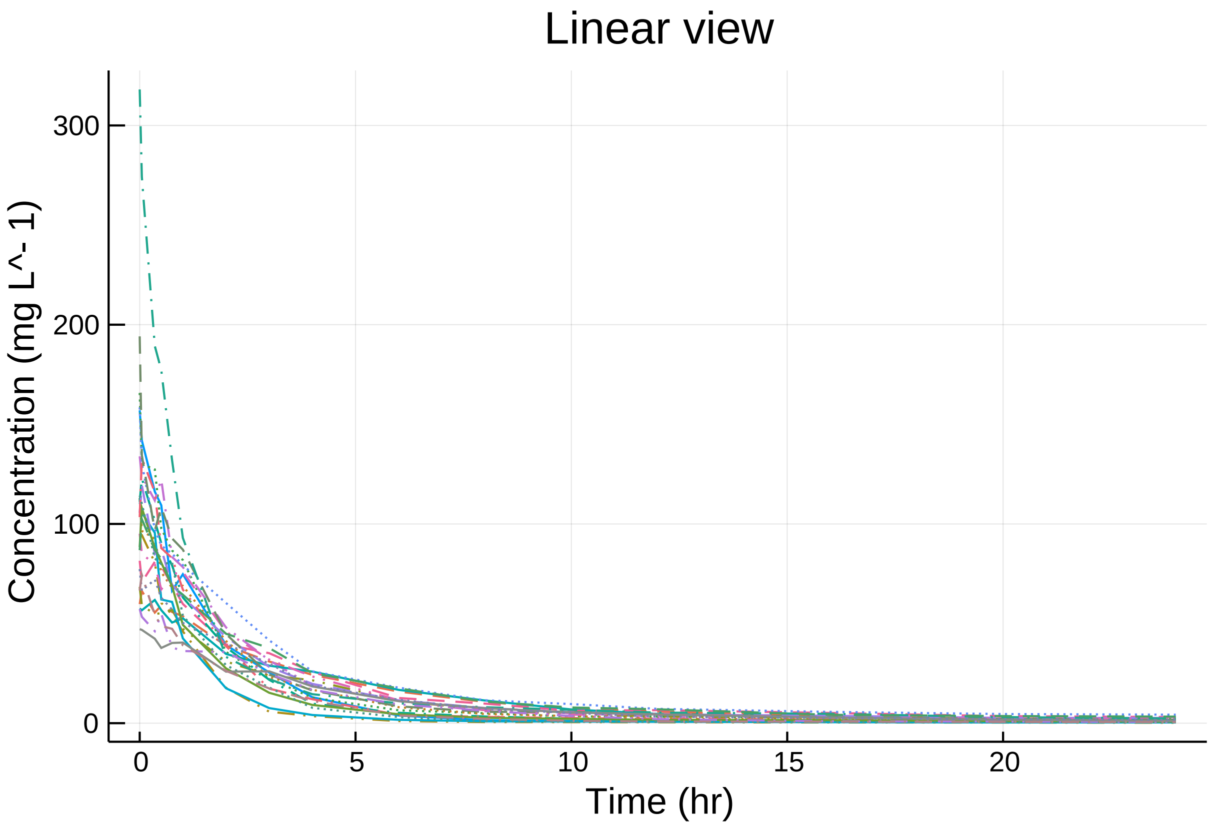
To generate linear and log-linear plots, one can do:

```
using Plots # load the plotting library
plot(pop)
```



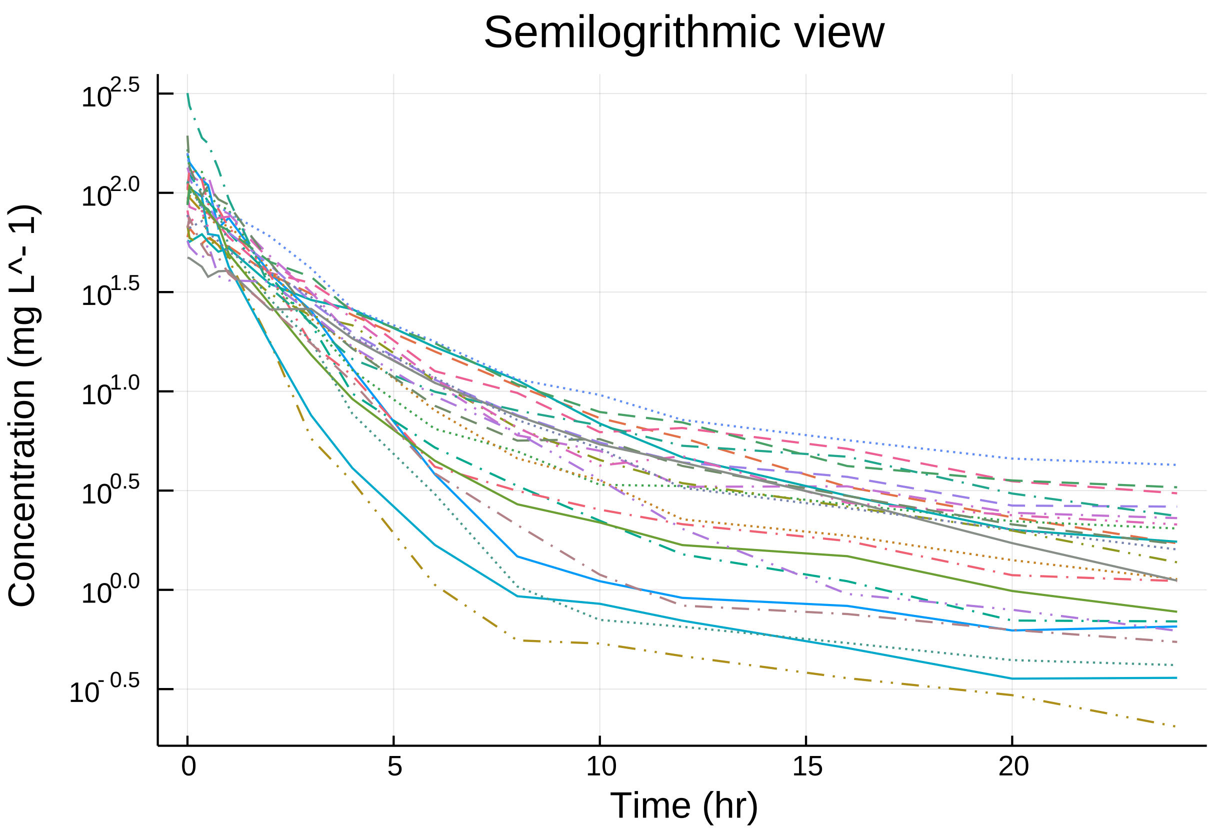
to only generate the linear plot:

```
plot(pop, loglinear=false)
```



Similarly, to generate log-linear plot:

```
plot(pop, linear=false)
```



To calculate all NCA quantities, one can do

```
report = NCAReport(pop)
```

```
NCAReport
  keys: Symbol[:id, :lambda_z, :half_life, :tmax, :cmax, :c0, :clast, :clas
t_pred, :auclast, :tlast, :aucinf_obs, :vz_obs, :cl_obs, :aucinf_pred, :vz_
pred, :cl_pred, :vss_obs, :vss_pred, :tmin, :cmin, :cminss, :cmax_d, :aucla
st_d, :aucinf_d_obs, :auc_extrap_obs, :auc_back_extrap_obs, :aucinf_d_pred,
 :auc_extrap_pred, :auc_back_extrap_pred, :aumclast, :aumcinf_obs, :aumc_ex
trap_obs, :aumcinf_pred, :aumc_extrap_pred, :mrtlast, :mrtinf_obs, :mrtinf_
pred, :n_samples, :rsq, :rsq_adjusted, :corr_xy, :no_points_lambda_z, :lamb
da_z_intercept, :lambda_z_lower, :lambda_z_upper, :span, :route]
```

The `NCAReport` object holds all quantities, and one can call `NCA.to_dataframe` to get a `DataFrame` object.

```
NCA.to_dataframe(report)
```

	id	doseamt	lambda_z	half_life	tmax	cmax	c0	
	Int64	Quantity	Quantity	Quantity	Quantity	Quantity	Quantity	
1	1	5000.0 mg	0.0387671 hr^{-1}	17.8798 hr	0.0 hr	157.021 mg L^{-1}	157.021 mg L^{-1}	0.
2	2	5000.0 mg	0.0817121 hr^{-1}	8.48279 hr	0.05 hr	66.354 mg L^{-1}	59.7702 mg L^{-1}	1
3	3	5000.0 mg	0.0397477 hr^{-1}	17.4387 hr	0.0 hr	165.733 mg L^{-1}	165.733 mg L^{-1}	2
4	4	5000.0 mg	0.0581041 hr^{-1}	11.9294 hr	0.0 hr	133.911 mg L^{-1}	133.911 mg L^{-1}	2
5	5	5000.0 mg	0.0662631 hr^{-1}	10.4605 hr	0.0 hr	94.9497 mg L^{-1}	94.9497 mg L^{-1}	0.
6	6	5000.0 mg	0.146807 hr^{-1}	4.72149 hr	0.35 hr	61.846 mg L^{-1}	57.5621 mg L^{-1}	1
7	7	5000.0 mg	0.0662241 hr^{-1}	10.4667 hr	0.0 hr	81.5067 mg L^{-1}	81.5067 mg L^{-1}	3
8	8	5000.0 mg	0.0591092 hr^{-1}	11.7266 hr	0.0 hr	113.0 mg L^{-1}	113.0 mg L^{-1}	1
9	9	5000.0 mg	0.112094 hr^{-1}	6.18365 hr	0.05 hr	122.842 mg L^{-1}	111.091 mg L^{-1}	0.
10	10	5000.0 mg	0.0758397 hr^{-1}	9.13964 hr	0.0 hr	68.4081 mg L^{-1}	68.4081 mg L^{-1}	1
11	11	5000.0 mg	0.0663746 hr^{-1}	10.443 hr	0.0 hr	110.864 mg L^{-1}	110.864 mg L^{-1}	0.
12	12	5000.0 mg	0.0541511 hr^{-1}	12.8002 hr	0.05 hr	119.39 mg L^{-1}	105.605 mg L^{-1}	2
13	13	5000.0 mg	0.0445008 hr^{-1}	15.576 hr	0.0 hr	158.479 mg L^{-1}	158.479 mg L^{-1}	4
14	14	5000.0 mg	0.0664999 hr^{-1}	10.4233 hr	0.05 hr	132.907 mg L^{-1}	103.412 mg L^{-1}	1
15	15	5000.0 mg	0.0314503 hr^{-1}	22.0395 hr	0.0 hr	95.0664 mg L^{-1}	95.0664 mg L^{-1}	2
16	16	5000.0 mg	0.0756395 hr^{-1}	9.16382 hr	0.05 hr	107.966 mg L^{-1}	88.4042 mg L^{-1}	0.
17	17	5000.0 mg	0.0763209 hr^{-1}	9.08201 hr	0.0 hr	194.133 mg L^{-1}	194.133 mg L^{-1}	1
18	18	5000.0 mg	0.0396862 hr^{-1}	17.4657 hr	0.0 hr	114.244 mg L^{-1}	114.244 mg L^{-1}	0.
19	19	5000.0 mg	0.0778716 hr^{-1}	8.90116 hr	0.0 hr	318.072 mg L^{-1}	318.072 mg L^{-1}	2
20	20	5000.0 mg	0.0534663 hr^{-1}	12.9642 hr	0.0 hr	57.2367 mg L^{-1}	57.2367 mg L^{-1}	0.
21	21	5000.0 mg	0.113931 hr^{-1}	6.08392 hr	0.0 hr	47.1919 mg L^{-1}	47.1919 mg L^{-1}	1
22	22	5000.0 mg	0.0307751 hr^{-1}	22.523 hr	0.05 hr	102.389 mg L^{-1}	86.8349 mg L^{-1}	3
23	23	5000.0 mg	0.0597314 hr^{-1}	11.6044 hr	0.0 hr	77.3496 mg L^{-1}	77.3496 mg L^{-1}	1
24	24	5000.0 mg	0.0404818 hr^{-1}	17.1224 hr	0.05 hr	74.3649 mg L^{-1}	66.5963 mg L^{-1}	0.

4 Multiple doses

The interface of doing NCA with multiple doses is the same as doing single dose NCA. To load the data with multiple doses, one can do

```
multiple_doses_file = Pumas.example_nmtran_data("nca_test_data/dapa_IV_ORAL")
mdata = CSV.read(multiple_doses_file)

timeu = u"hr"
```

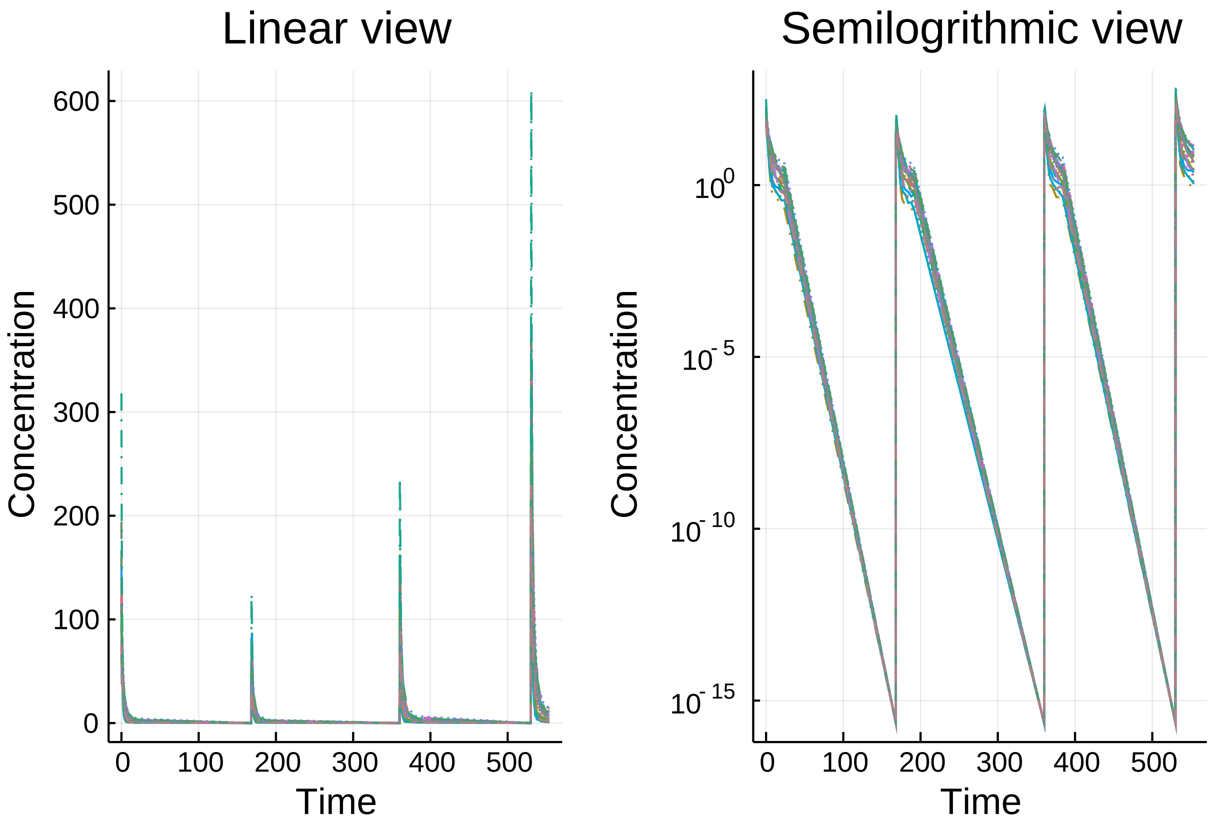
```
concu = u"mg/L"
amtu  = u"mg"
mdata.id = mdata.ID
mdata.time = mdata.TIME
mdata.conc = mdata.COBS
mdata.amt = mdata.AMT
mdata.route = replace(mdata.FORMULATION, "IV"=>"iv", "ORAL"=>"ev")
mdata.occasion = mdata.OCC
mpop = read_nca(mdata, timeu=timeu, concu=concu, amtu=amtu)
```

```
NCAPopulation (24 subjects):
  ID: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 2
0, 21, 22, 23, 24]
    concentration: mg L^-1
    time:          hr
    auc:           mg hr L^-1
    aumc:          mg hr^2 L^-1
    λz:            hr^-1
    dose:          mg
```

Note that to read multiple doses files, in addition to single dose inputs, one also needs to provide the `occasion` column.

To plot:

```
plot(mpop)
```



To compute AUC and λz:

```
NCA.auc(mpop)
```

	id	occasion	auc
	Int64	Int64	Quantity
1	1	1	263.793 mg hr L^{-1}
2	1	2	202.165 mg hr L^{-1}
3	1	3	421.956 mg hr L^{-1}
4	1	4	1049.24 mg hr L^{-1}
5	2	1	323.253 mg hr L^{-1}
6	2	2	252.969 mg hr L^{-1}
7	2	3	491.522 mg hr L^{-1}
8	2	4	1223.64 mg hr L^{-1}
9	3	1	339.848 mg hr L^{-1}
10	3	2	245.484 mg hr L^{-1}
11	3	3	482.17 mg hr L^{-1}
12	3	4	1189.24 mg hr L^{-1}
13	4	1	373.361 mg hr L^{-1}
14	4	2	302.52 mg hr L^{-1}
15	4	3	583.329 mg hr L^{-1}
16	4	4	1354.45 mg hr L^{-1}
17	5	1	132.145 mg hr L^{-1}
18	5	2	106.014 mg hr L^{-1}
19	5	3	197.069 mg hr L^{-1}
20	5	4	484.693 mg hr L^{-1}
21	6	1	303.86 mg hr L^{-1}
22	6	2	251.517 mg hr L^{-1}
23	6	3	462.855 mg hr L^{-1}
24	6	4	1122.27 mg hr L^{-1}
25	7	1	380.275 mg hr L^{-1}
26	7	2	286.747 mg hr L^{-1}
27	7	3	627.525 mg hr L^{-1}
28	7	4	1965.43 mg hr L^{-1}
29	8	1	279.126 mg hr L^{-1}
30	8	2	199.735 mg hr L^{-1}
31	8	3	412.669 mg hr L^{-1}
32	8	4	947.268 mg hr L^{-1}
33	9	1	239.831 mg hr L^{-1}
34	9	2	169.627 mg hr L^{-1}
35	9	3	348.093 mg hr L^{-1}
36	9	4	879.198 mg hr L^{-1}
37	10	1	260.862 mg hr L^{-1}
38	10	2	194.393 mg hr L^{-1}
39	10	3	367.769 mg hr L^{-1}
40	10	4	1012.01 mg hr L^{-1}
41	11	1	146.864 mg hr L^{-1}
42	11	2	103.955 mg hr L^{-1}
43	11	3	196.418 mg hr L^{-1}
44	11	4	519.961 mg hr L^{-1}
45	12	1	359.489 mg hr L^{-1}
46	12	2	289.377 mg hr L^{-1}
47	12	3	511.423 mg hr L^{-1}
48	12	4	1320.47 mg hr L^{-1}
49	13	1	522.905 mg hr L^{-1}
50	13	2	366.313 mg hr L^{-1}
51	13	3	756.928 mg hr L^{-1}
52	13	4	1785.53 mg hr L^{-1}

To get a summary, we need to provide a reference dose. In this example, we are going to let the first dose be the reference dose.

```
rep = NCAReport(mpop, ithdose=1)
NCA.to_dataframe(rep)
```

	id	occasion	doseamt	lambda_z	half_life	tmax	tlag	cmax
	Int64	Int64	Quantity	Quantity	Quantity	Quantity	Quantit	Quantity
1	1	1	5000 mg	0.0387671 hr^-1	17.8798 hr	0.0 hr		157.021 mg L^-1
2	1	2	5000 mg	0.192171 hr^-1	3.60693 hr	1.0 hr	0.0 hr	86.3231 mg L^-1
3	1	3	10000 mg	0.0320026 hr^-1	21.6591 hr	0.75 hr	0.0 hr	161.098 mg L^-1
4	1	4	25000 mg	0.0244296 hr^-1	28.3732 hr	0.5 hr	0.0 hr	385.549 mg L^-1
5	2	1	5000 mg	0.0817121 hr^-1	8.48279 hr	0.05 hr		66.354 mg L^-1
6	2	2	5000 mg	0.102657 hr^-1	6.75204 hr	1.0 hr	0.0 hr	42.9419 mg L^-1
7	2	3	10000 mg	0.145337 hr^-1	4.76924 hr	1.0 hr	0.0 hr	87.8165 mg L^-1
8	2	4	25000 mg	0.0919952 hr^-1	7.5346 hr	0.5 hr	0.0 hr	192.247 mg L^-1
9	3	1	5000 mg	0.0397477 hr^-1	17.4387 hr	0.0 hr		165.733 mg L^-1
10	3	2	5000 mg	0.0478226 hr^-1	14.4941 hr	0.75 hr	0.0 hr	75.0977 mg L^-1
11	3	3	10000 mg	0.0578421 hr^-1	11.9834 hr	0.75 hr	0.0 hr	170.954 mg L^-1
12	3	4	25000 mg	0.0621692 hr^-1	11.1494 hr	0.75 hr	0.0 hr	369.544 mg L^-1
13	4	1	5000 mg	0.0581041 hr^-1	11.9294 hr	0.0 hr		133.911 mg L^-1
14	4	2	5000 mg	0.0566557 hr^-1	12.2344 hr	1.0 hr	0.0 hr	72.6184 mg L^-1
15	4	3	10000 mg	0.0518813 hr^-1	13.3602 hr	0.75 hr	0.0 hr	134.983 mg L^-1
16	4	4	25000 mg	0.0815239 hr^-1	8.50238 hr	0.5 hr	0.0 hr	375.488 mg L^-1
17	5	1	5000 mg	0.0662631 hr^-1	10.4605 hr	0.0 hr		94.9497 mg L^-1
18	5	2	5000 mg	0.0239926 hr^-1	28.89 hr	0.5 hr	0.0 hr	48.6987 mg L^-1
19	5	3	10000 mg	0.0664184 hr^-1	10.4361 hr	0.75 hr	0.0 hr	108.837 mg L^-1
20	5	4	25000 mg	0.0815214 hr^-1	8.50264 hr	0.35 hr	0.0 hr	258.304 mg L^-1
21	6	1	5000 mg	0.146807 hr^-1	4.72149 hr	0.35 hr		61.846 mg L^-1
22	6	2	5000 mg	0.068555 hr^-1	10.1108 hr	1.0 hr	0.0 hr	40.8591 mg L^-1
23	6	3	10000 mg	0.098923 hr^-1	7.00694 hr	0.5 hr	0.0 hr	75.5742 mg L^-1
24	6	4	25000 mg	0.144537 hr^-1	4.79564 hr	0.5 hr	0.0 hr	197.489 mg L^-1
25	7	1	5000 mg	0.0662241 hr^-1	10.4667 hr	0.0 hr		81.5067 mg L^-1
26	7	2	5000 mg	0.0586968 hr^-1	11.809 hr	0.5 hr	0.0 hr	52.4783 mg L^-1
27	7	3	10000 mg	0.0401077 hr^-1	17.2821 hr	0.75 hr	0.0 hr	109.742 mg L^-1
28	7	4	25000 mg	0.0241078 hr^-1	28.7519 hr	1.0 hr	0.0 hr	297.977 mg L^-1
29	8	1	5000 mg	0.0591092 hr^-1	11.7266 hr	0.0 hr		113.0 mg L^-1
30	8	2	5000 mg	0.0559283 hr^-1	12.3935 hr	0.75 hr	0.0 hr	55.3256 mg L^-1
31	8	3	10000 mg	0.0440135 hr^-1	15.7485 hr	0.75 hr	0.0 hr	111.597 mg L^-1
32	8	4	25000 mg	0.0900523 hr^-1	7.69717 hr	0.5 hr	0.0 hr	287.159 mg L^-1
33	9	1	5000 mg	0.112094 hr^-1	6.18365 hr	0.05 hr		122.842 mg L^-1
34	9	2	5000 mg	0.0678077 hr^-1	10.2223 hr	0.5 hr	0.0 hr	51.359 mg L^-1
35	9	3	10000 mg	0.0840572 hr^-1	8.24613 hr	1.0 hr	0.0 hr	120.764 mg L^-1
36	9	4	25000 mg	0.0592211 hr^-1	11.7044 hr	0.75 hr	0.0 hr	328.378 mg L^-1
37	10	1	5000 mg	0.0758397 hr^-1	9.13964 hr	0.0 hr		68.4081 mg L^-1
38	10	2	5000 mg	0.0793964 hr^-1	8.73021 hr	1.0 hr	0.0 hr	42.7339 mg L^-1
39	10	3	10000 mg	0.136161 hr^-1	5.09063 hr	0.5 hr	0.0 hr	83.3259 mg L^-1
40	10	4	25000 mg	0.0644552 hr^-1	10.7539 hr	0.75 hr	0.0 hr	179.225 mg L^-1
41	11	1	5000 mg	0.0663746 hr^-1	10.443 hr	0.0 hr		110.864 mg L^-1
42	11	2	5000 mg	0.0901918 hr^-1	7.68526 hr	0.5 hr	0.0 hr	51.6522 mg L^-1
43	11	3	10000 mg	0.0657487 hr^-1	10.5424 hr	0.5 hr	0.0 hr	86.9005 mg L^-1
44	11	4	25000 mg	0.0670837 hr^-1	10.3326 hr	0.5 hr	0.0 hr	227.847 mg L^-1
45	12	1	5000 mg	0.0541511 hr^-1	12.8002 hr	0.05 hr		119.39 mg L^-1
46	12	2	5000 mg	0.0465747 hr^-1	14.8825 hr	0.75 hr	0.0 hr	68.0056 mg L^-1
47	12	3	10000 mg	0.0643446 hr^-1	10.7724 hr	0.75 hr	0.0 hr	132.711 mg L^-1
48	12	4	25000 mg	0.0613251 hr^-1	11.3028 hr	1.0 hr	0.0 hr	334.842 mg L^-1
49	13	1	5000 mg	0.0445008 hr^-1	15.576 hr	0.0 hr		158.479 mg L^-1
50	13	2	5000 mg	0.0620317 hr^-1	11.1741 hr	0.75 hr	0.0 hr	79.6555 mg L^-1