

Comparison of non-compartmental analysis results between PKNCA, Pumas and Phoenix WinNonlin

Yingbo Ma^{1,2}, Beatriz Guglieri-Lopez¹, Joga Gobburu¹, William Denney³, Vijay Ivaturi¹

¹Center for Translational Medicine, University of Maryland School of Pharmacy, Baltimore, MD, USA; ²JuliaLab, MIT, MA, USA; ³Human Predictions, LLC, Cambridge, MA, USA

Introduction

Non-Compartmental Analysis (NCA) is critical to understanding pharmacokinetic characteristics of drugs required for Phase 1 Data Review, IND, NDA and IB input and as required for other regulatory interactions and internal decision making contexts.

There are many existing tools to perform NCA analysis that can be used for regulatory filing. In this work, we set out to compare the numerical accuracy of the obtained results across three software

1. PKNCA – An R based NCA tool developed by Human Predictions
2. Pumas NCA - A Julia Language based implementation of NCA in Pumas
3. Phoenix WinNonlin – a commonly used Windows-based software for NCA

Both PKNCA and Pumas NCA are cross platform whereas Phoenix runs only on Windows.

Methods

Simulations were performed in R with scenarios consisting of 1-, 2-, and 3-compartment models with typical parameters and ± 4 -fold from those typical parameters on the ratio of absorption rate (K_a) to elimination rate (K_{el}), ratio of peripheral volume of distribution 1 and 2 (V_{p1} and V_{p2}) to central volume of distribution (V_c), intercompartmental clearance between V_c and V_{p1} or V_{p2} (Q_{cp1} and Q_{cp2}), and ratio of clearance (CL) to V_c all models, as the parameters apply; with and without target-mediated drug disposition (TMDD); and oral and intravascular bolus dosing. All models were simulated with $\sim 4\%$, 10% , and 20% proportional residual error. Each model was simulated with 6 subjects. This yielded a total of 13104 scenarios and 78624 subjects simulated.

Each subject was then grouped with all other subjects in its simulation scenario, and 5, 10, and 20% of concentration measurements were set to below the limit of quantification (LOQ). NCA was to be performed on each of those LOQ scenarios in PKNCA, Pumas, and Phoenix (a total of 707616 NCA intervals with calculations). Comparisons were made between the results of those NCA calculations performed on a single machine

Results

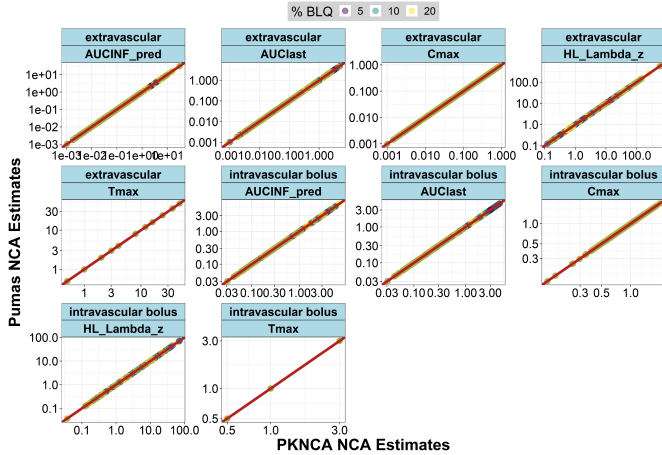
Five parameters, were chosen as the metrics for comparison as they included both observed and derived parameters:

- AUClast
- Cmax
- Tmax
- Half-life
- AUCinf(pred)

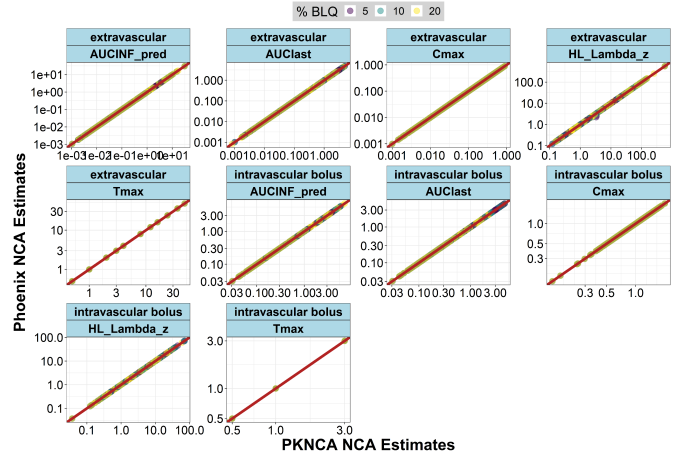
Eighteen percent of subjects were randomly selected from the 13104 scenarios to form a subset of 2367 subjects. This smaller subset was used to perform the NCA calculations across the three software while ensuring to maintain the same default options. All analysis were conducted on the same laptop.

Figures 1, 2 and 3 show that the derived NCA parameters are similar across all three software. Table 1 shows the total percent difference of PKNCA and Pumas with respect to Phoenix if there were any.

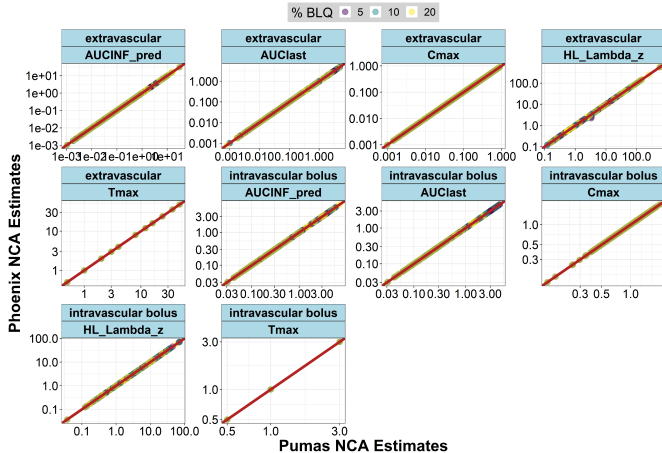
Comparison of PKNCA and Pumas parameter estimates



Comparison of PKNCA and Phoenix parameter estimates



Comparison of Pumas and Phoenix parameter estimates



The table below shows the fraction of results different between software. PKNCA and Pumas had near-complete agreement (Figure 3) and hence are not shown.

Parameter	Pumas/Phoenix difference			PKNCA/Phoenix difference		
	$\geq 0.1\%$	$\geq 1\%$	$\geq 5\%$	$\geq 0.1\%$	$\geq 1\%$	$\geq 5\%$
AUC _{∞,pred}	0.0	0.254	0.134	0.0	0.0	0.0
AUC _{last}	0.0	0.0	0.0	0.0	0.0	0.0
C _{max}	0.0	0.0	0.0	0.0	0.0	0.0
Half-life	0.007	0.07	0.043	0.043	0.043	0.043
T _{max}	0.0	0.0	0.0	0.0	0.0	0.0

Conclusions and Discussions

- Results from all three software for the key NCA parameters match.
- Most results matched within $\pm 0.1\%$ between all software.
- The difference between PKNCA /Pumas and Phoenix in half-life appears to be the result of PKNCA/Pumas selecting the best fit first and then filtering for decreasing slope while Phoenix first considers only consecutive sets of points that generate a descending slope and then selects the final set of points with the best regression adjusted R squared.
- None of the different results would be reported in a typical reporting workflow as all r^2 values were < 0.7 .

We extend our sincere thanks to Ana Henry of Certara for her help in ensuring the correct analysis in Phoenix