mariev

{VALTOOLS}

SOFTWARE VALIDATION SEPARATE FROM DEVELOPMENT FOR PERFORMANCE QUALIFICATION TESTING

SCHARP at FRED HUTCH

Marie Vendettuoli
Senior Statistical Programmer

ACKNOWLEDGEMENTS

Statistical Center for HIV/AIDS Research and Prevention SCHARP at FRED HUTCH

Ellis Hughes
Rafael Kuttner
Kate Ostbye
Paul Stutzman
Anthony Williams
+ Pdata Team

VISC Vaccine Immunology Statistical Center

HIV VACCINE TRIALS NETWORK

phuse Data Visualisation & Open Source Technology phuse.global

Peyman Eshghi
Maya Gans
Eli Miller
Mike Stackhouse

Hanming Tu
+ R Package Validation Team

TABLE OF CONTENTS

SCHARP at FRED HUTCH

GOALS OF SOFTWARE VALIDATION

"Truth" vs "A Regulatory Requirement"

GOALS OF SOFTWARE VALIDATION

Truth-seeking

- How do I know this software is “fit for purpose”? Does this software play nicely with existing organizational infrastructure and resources?
- Have the developers considered the edge cases that are important to me? Is this performance reproducible?
- How much effort will this *really* take? What happens when I make an update?

GOALS OF SOFTWARE VALIDATION

Truth-seeking

- How do I know this software is "fit for purpose"? Does this software play nicely with existing organizational infrastructure and resources?
- Have the developers considered the edge cases that are important to me? Is this performance reproducible?
- How much effort will this *really* take? What happens when I make an update?

GOALS OF SOFTWARE VALIDATION

Fulfilling Regulatory Requirements

General Principles of Software Validation; Final Guidance for Industry and FDA Staff

FDA considers software validation to be "confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled."

https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation

GOALS OF SOFTWARE VALIDATION

Fulfilling Regulatory Requirements

General Principles of Software Validation; Final Guidance for Industry and FDA Staff

Fit for purpose

ers software validation to be "confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled."

https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation

GOALS OF SOFTWARE VALIDATION

Fulfilling Regulatory Requirements

Reproducibility

General Principles of Software Validation; Final Guidance for Industry and FDA Staff

FDA considers software validation to be "confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled."

https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation

GOALS OF SOFTWARE VALIDATION

Fulfilling Regulatory Requirements

Effort

General Principles of Software Validation; Final Guidance for Industry and FDA Staff

FDA considers software validation to be "confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled."

https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation

WHY IS THIS DISTINCTION IMPORTANT?

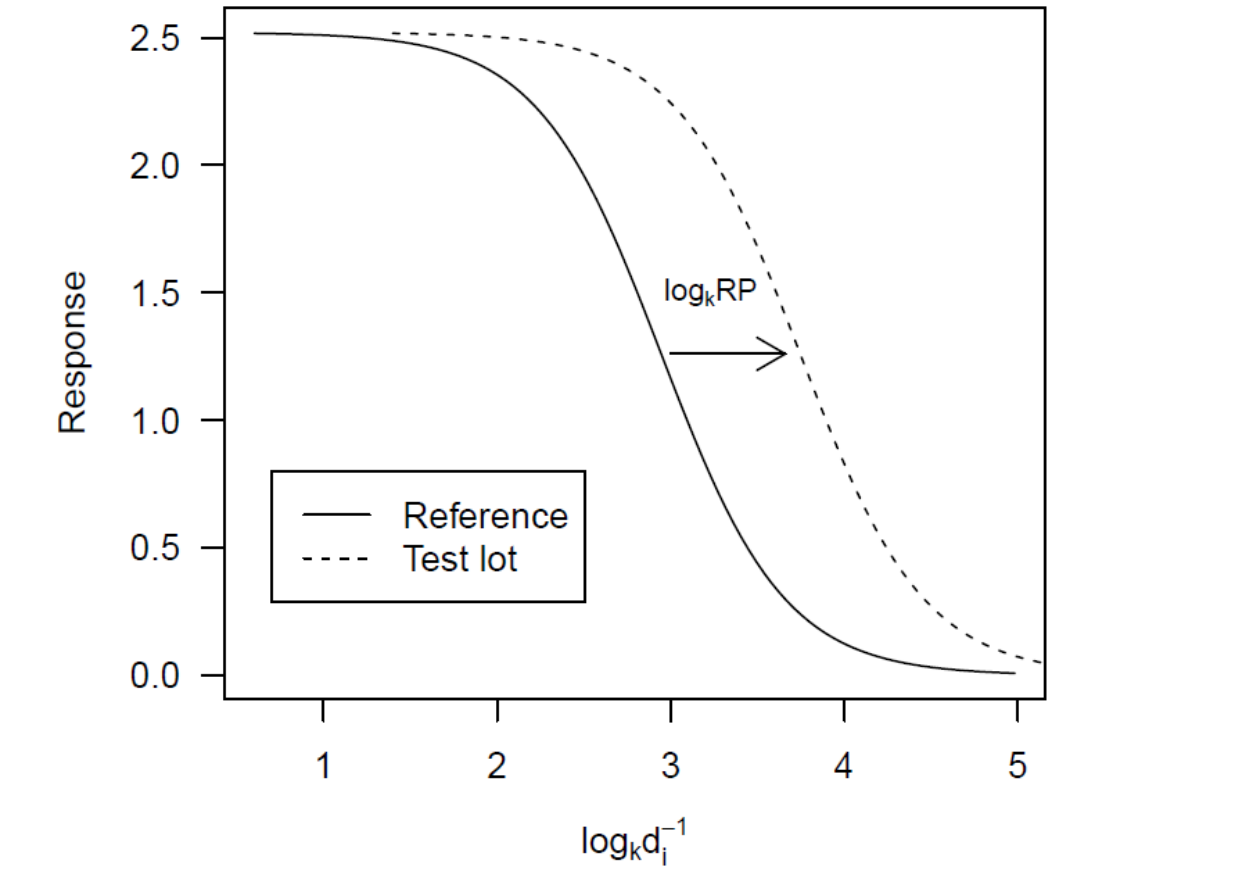
Example from USDA APHIS Center for Veterinary Biologics (CVB)
Estimating relative potency in vaccines

SCHARP at FRED HUTCH

Relative Potency – An estimate of performance difference between lots

$$\log_k RP = \log_k C_{\text{test lot}} - \log_k C_{\text{reference}} = \log_k \left(\frac{C_{\text{test lot}}}{C_{\text{reference}}} \right)$$

$$y = D + \frac{A - D}{1 + \left(\frac{x}{C}\right)^B}.$$



Siev D. Interpretation and estimation of relative potency in vaccines. J Immunol Methods. 1997 Oct 27;208(2):131-9. doi: 10.1016/s0022-1759(97)00134-8. PMID: 9433468.

Tong, C. and Vendettuoli, M. (2017) USDA APHIS STATWI0005.01 - Logistic Curve Parameterizations in Commercial Software. https://www.aphis.usda.gov/animal_health/vet_biologics/publications/STATWI0005.pdf

$$\log_k RP = \log_k C_{\text{test lot}} - \log_k C_{\text{reference}} = \log_k \left(\frac{C_{\text{test lot}}}{C_{\text{reference}}} \right)$$

Lower asymptote

Theoretically should be zero!!

Upper asymptote

$$y = D + \frac{A - D}{1 + \left(\frac{x}{C}\right)^B}.$$

Response

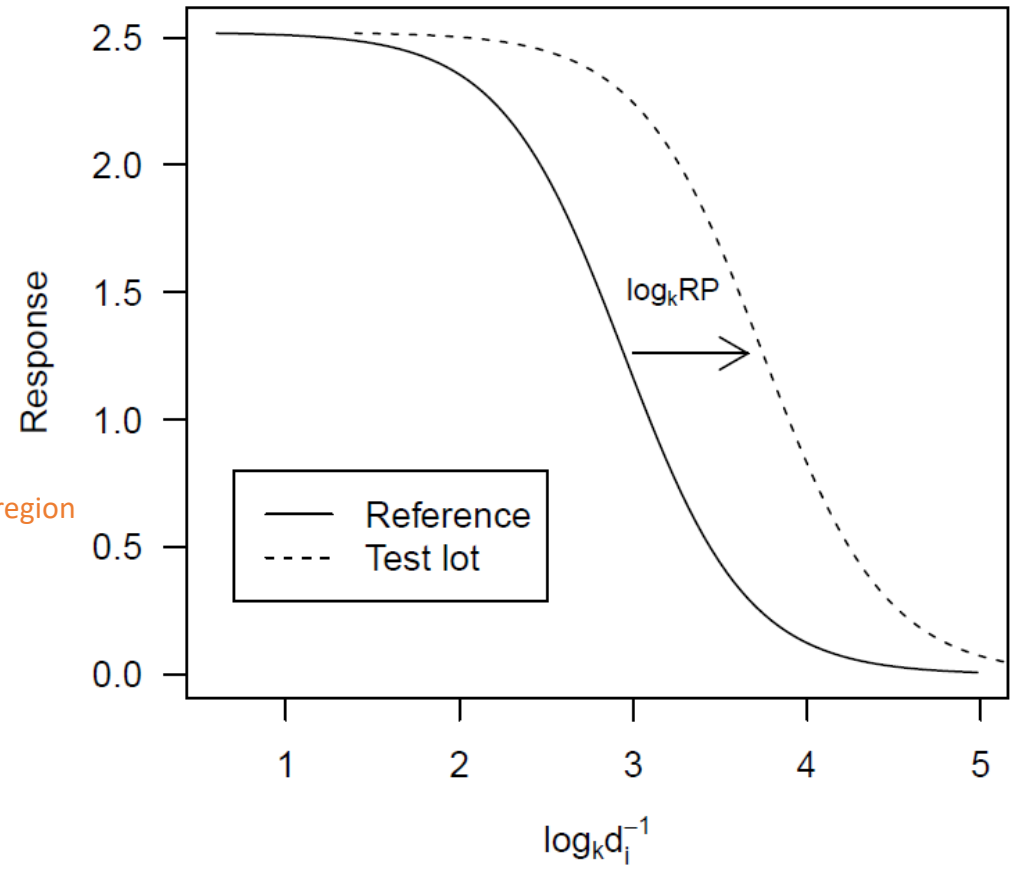
ELISA OD

Scale factor

Sign = direction of pseudo-linear region

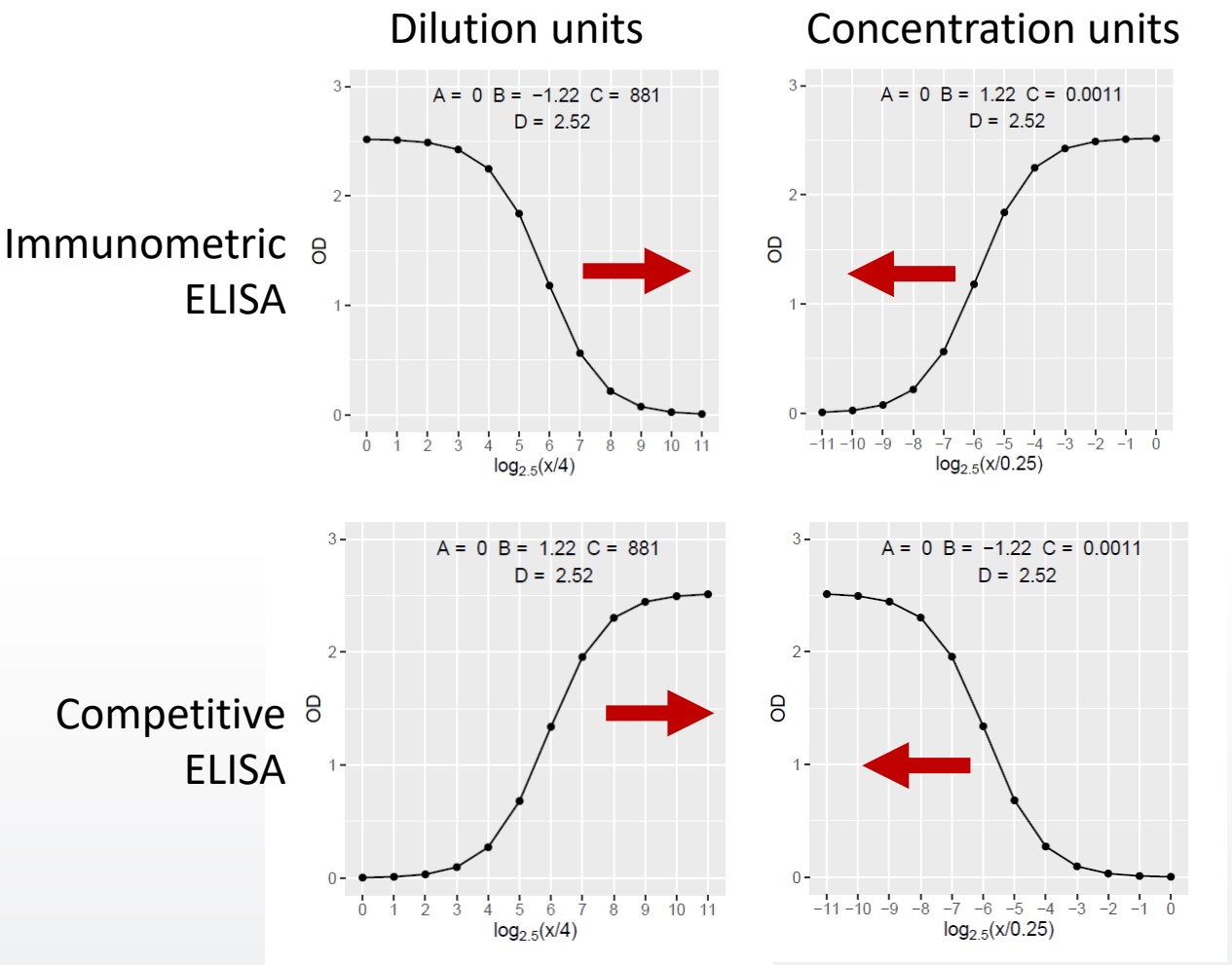
Serial dilution

Location parameter



Siev D. Interpretation and estimation of relative potency in vaccines. J Immunol Methods. 1997 Oct 27;208(2):131-9. doi: 10.1016/s0022-1759(97)00134-8. PMID: 9433468.

Tong, C. and Vendettuoli, M. (2017) USDA APHIS STATWI0005.01 - Logistic Curve Parameterizations in Commercial Software. https://www.aphis.usda.gov/animal_health/vet_biologics/publications/STATWI0005.pdf



Response type	Units of x	Constraint	Sign of B	Potency increases
immunometric	dilution	$A = 0$	$-$	to the right
competitive	dilution	$A = 0$	$+$	to the right
immunometric	concentration	$A = 0$	$+$	to the left
competitive	concentration	$A = 0$	$-$	to the left

Table 1: Summary of conventions for 4PL constraints and sign of B for enforcing a 3PL fit.

Tong, C. and Vendettuoli, M. (2017) USDA APHIS STATWI0005.01 - Logistic Curve Parameterizations in Commercial Software. https://www.aphis.usda.gov/animal_health/vet_biologics/publications/STATWI0005.pdf

Response type	Units of x	Constraint	Sign of B	Potency increases
immunometric	dilution	$A = 0$	$-$	to the right
competitive	dilution	$A = 0$	$+$	to the right
immunometric	concentration	$A = 0$	$+$	to the left
competitive	concentration	$A = 0$	$-$	to the left

Table 1: Summary of conventions for 4PL constraints and sign of B for enforcing a 3PL fit.

USDA APHIS Center for Veterinary Biologics' internal practice 1997 - present

Lower asymptote

Software manufacturer implementation 2006 - present

Response type	Units of x	Constraint in Gen5	Sign of B	Potency increases
immunometric	dilution	$D = 0$	$+$	to the right
competitive	dilution	$A = 0$	$+$	to the right
immunometric	concentration	$A = 0$	$+$	to the left
competitive	concentration	$D = 0$	$+$	to the left

Table 2: Summary of conventions for 4PL constraints and sign of B for enforcing a 3PL fit in Biotek Gen5 software.

Tong, C. and Vendettuoli, M. (2017) USDA APHIS STATWI0005.01 - Logistic Curve Parameterizations in Commercial Software. https://www.aphis.usda.gov/animal_health/vet_biologics/publications/STATWI0005.pdf

SOFTWARE VALIDATION

out of scope (for this talk)

- Validation and verification of data
- Decision-making component of identifying requirements, risk assessment
- Unit/regression/integration testing

SOFTWARE VALIDATION AS PROCESS

Define requirements

Define tests

Execute tests

Document observations

(repeat)

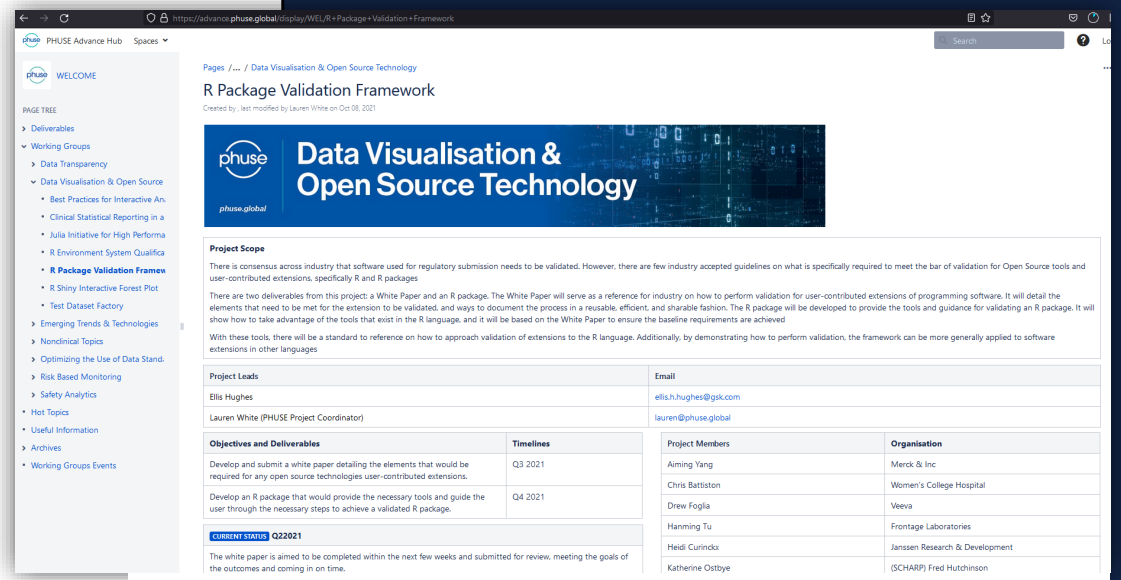
SOFTWARE VALIDATION AS PROCESS

Define requirements Define tests	*Fit for purpose*
Execute tests	*Reproducibility*
Document observations	*Effort*

R PACKAGE VALIDATION FRAMEWORK

Structuring the process for R

- Define elements of validation
  - Requirements
  - Test cases (optional)
  - Test code
  - Reports
- Specify a common organization of validation elements
- Process for evaluating the test code and generating reports
- System for sharing of validated code and reports
- Common approach for developers and useRs



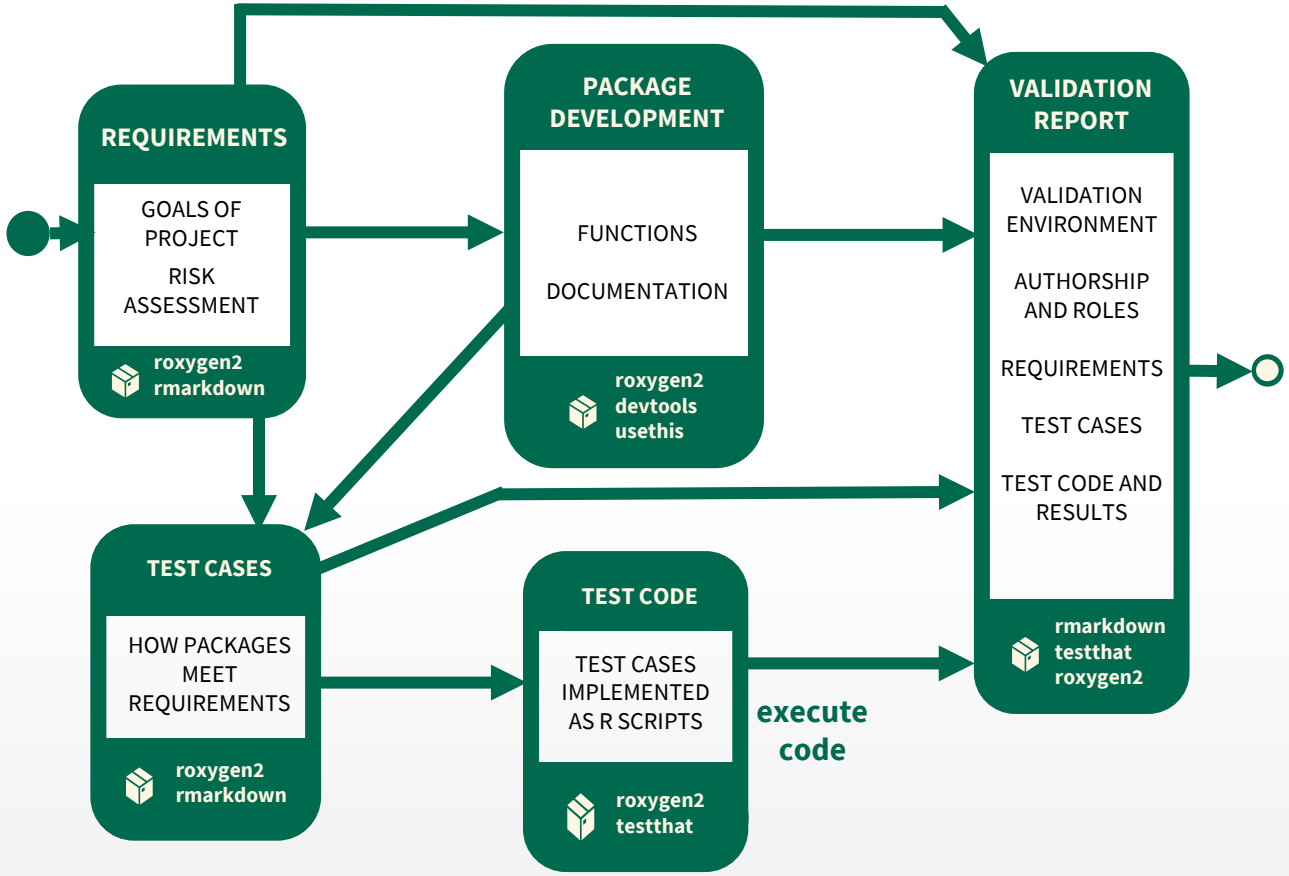
https://advance.phuse.global/display/WEL/R+Package+Validation+Framework

Statistical Center for HIV/AIDS Research and Prevention
SCHARP
at FRED HUTCH

R VALIDATION FRAMEWORK

structuring the process for R

```
.
+-- DESCRIPTION
+-- example.package.Rproj
+-- man
+-- NAMESPACE
+-- R
\-- vignettes
    +-- validation
    |   +-- change_log.md
    |   +-- requirements
    |   +-- test_cases
    |   +-- test_code
    |   \-- validation.yml
    \-- validation.Rmd
```



PHUSE (2021) R Package Validation Framework

INTRO TO {VALTOOLS}

https://github.com/phuse-org/valtools

Nov 05
10:00-13:00 ET

R Package Validation Framework

Hosted by Ellis Hughes (Fred Hutch) & Marie Vendettuoli (Fred Hutch)

Workshop Filled

INTRO TO {VALTOOLS}

```
.
+-- DESCRIPTION
+-- example.package.Rproj
+-- man
+-- NAMESPACE
+-- R
\-- vignettes
    +-- validation
    |   +-- change_log.md
    |   +-- requirements
    |   +-- test_cases
    |   +-- test_code
    |   \-- validation.yml
    \-- validation.Rmd
```

Start here!

Initiate framework and validation elements from templates

Manage report content

Execute validation in multiple modes

- Separate from development

- From source code
- At installation
- For distribution

- After installation

INTRO TO {VALTOOLS}

	No validation elements in installed package	Requires access to package source code that includes validation elements			Installed package must have validation elements
Choose validation location	separate from package	from source code	at installation	for distribution	after installation
Run the validation report	vt_validate_report()	vt_validate_source()	vt_validate_install()	vt_validate_build()	vt_validate_installed_package()
Look for output	`|-validation/` `|  |- report_v[x].pdf`	`|-inst/` `|  |- report_v[x].pdf`	`system.file("validation", "report_v[X].pdf", package = "PACKAGE")`	1 `|-inst/` `|  |- report_v[x].pdf` 2 `package_[X].tar.gz`	In designated location
What environment was used?	Current workspace, no changes introduced	Source code is installed to temp workspace; report is added to package skeleton	Package is installed to current workspace while rendering report	Source code is installed to temp workspace. Report is added to package skeleton and tarball.	Current workspace, no changes introduced

INTRO TO {VALTOOLS}

README.md

R-CMD-check passing | codecov 95% | License MIT

valtools

valtools helps automate the validation of R packages used in clinical research and drug development: It provides useful templates and helper functions for tasks that arise during project set up and development of the validation framework.

This package is created by the R Package Validation Framework PHUSE Working Group.

For background on the `R Package Validation Framework`, watch this presentation from RStudio::Global Pharma X Session.

Installation

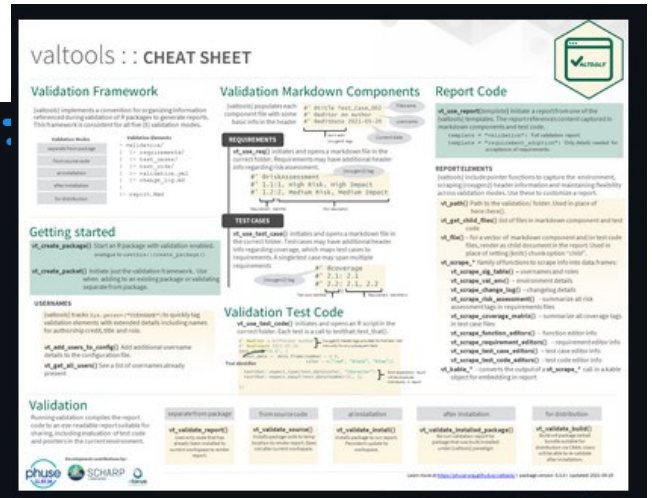
{valtools} is under active development. To get the latest version, install from github:

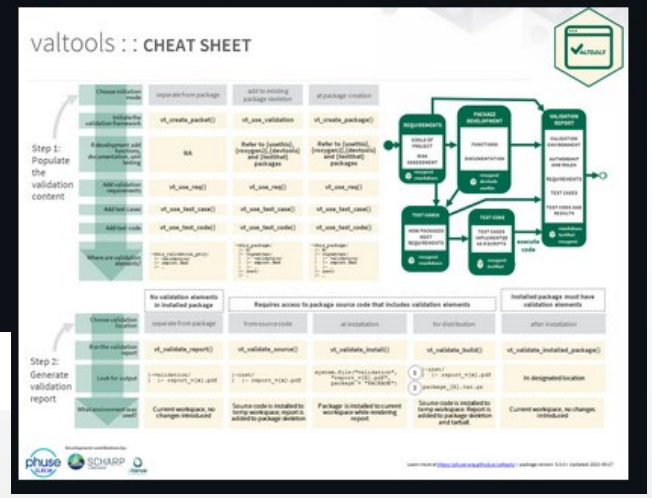
```
devtools::install_github("phuse-org/valtools")
```

Usage

{valtools} is intended to be a bolt-on addition to the standard package development process followed by usethis and devtools. All {valtools} functions start with `vt` followed by a verb/descriptor to make them easy to follow. The most common usage for valtools is to add elements required for validation under the R Package Validation Framework and facilitating validation. See the `Starting New Validation Package using {valtools}` vignette for more details.

https://github.com/phuse-org/valtools





PERFORMANCE QUALIFICATION (PQ) TESTING

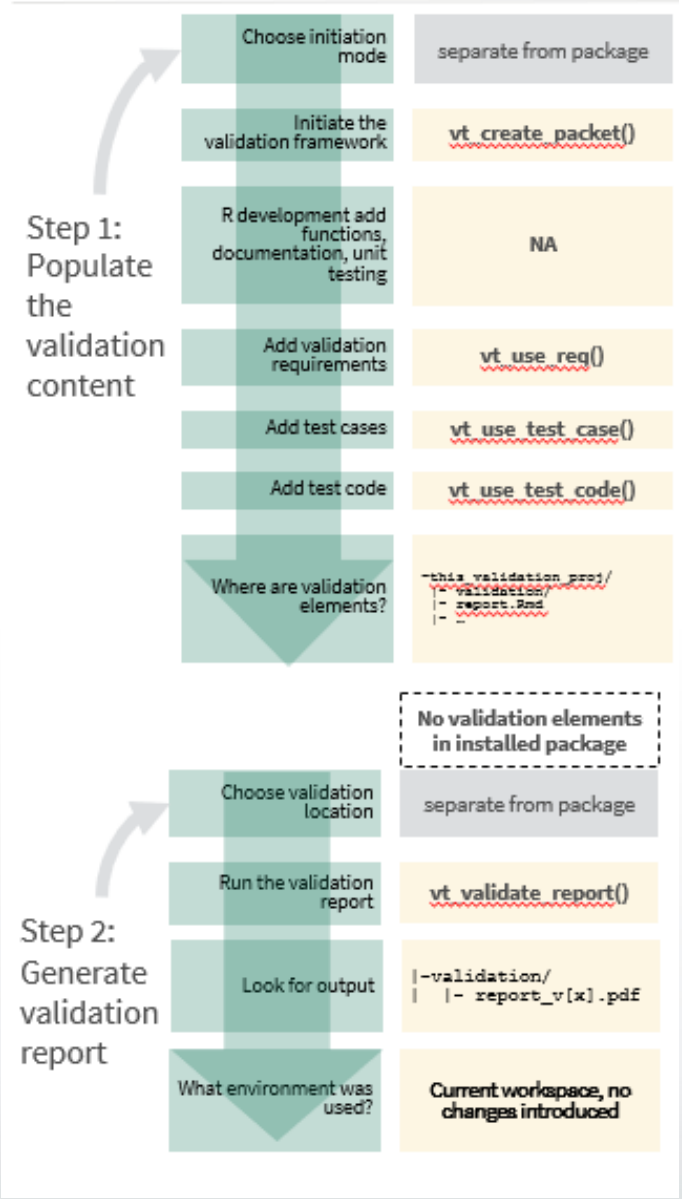
- Checks that the system performs as expected under real-world conditions
- Happens after installation/setup of system
- Evaluates the whole system, not just individual components
- May be repeated at specific intervals:
  - Periodic, e.g. daily, weekly
  - At startup
  - After a system change e.g. updating an R package

PERFORMANCE QUALIFICATION (PQ) TESTING

why {valtools} ?

```
+-- validation
|   +-- change_log.md
|   +-- requirements
|   +-- test_cases
|   +-- test_code
|   \-- validation.yml
\-- validation.Rmd
```

- Testing within an R environment
- Re-executable and extensible
- Similar to previous validation work



PQ TESTING @ SCHARP

Questions for consideration

- What is the default state of the working environment? Is this expected?
- Do we get the same results when switching environments, e.g. different servers or operating systems?
- What custom options are we setting, specific to our organization?
- For workflows that depend on functions from multiple packages, do we observe unexpected interaction?

- How do we communicate expectations between stakeholder groups?
- How does validation code evolve over time? Is this easily traceable?

PQ TESTING @ SCHARP

How do we communicate expectations between stakeholder groups?

Requirement report

Validation Author

10/18/2021

1. Character variables from csv import are not coerced to factors.

Validation report

Validation Author

10/18/2021

Case A

Requirement

1. Character variables from csv import are not coerced to factors.

Test case

- Read in the file `validation/testdata/rpharma_pp.csv` and save as `this_pp`. Verify that the variable "species" in `this_pp` is of class `character`.

Test result

Test	Results	Pass/Fail
TC_A1.1	As expected	Pass

PQ TESTING @ SCHARP

Keep it DRY

Requirement report

Validation Author

10/18/2021

1. Character variables from csv import are not coerced to factors.

```
+-- validation
|    +-- requirements
|    |    +-- req1.md
|    |    +-- ...
|    +-- test_cases
|    |    +-- test_case1.md
|    |    +-- ...
|    +-- test_code
|    |    +-- test_code1.R
|    |    +-- ...
|    +-- ...
+ ...
```

Validation report

Validatic

10/18

Case A

Requirement

1. Character variables from csv import are not coerced to factors.

Test case

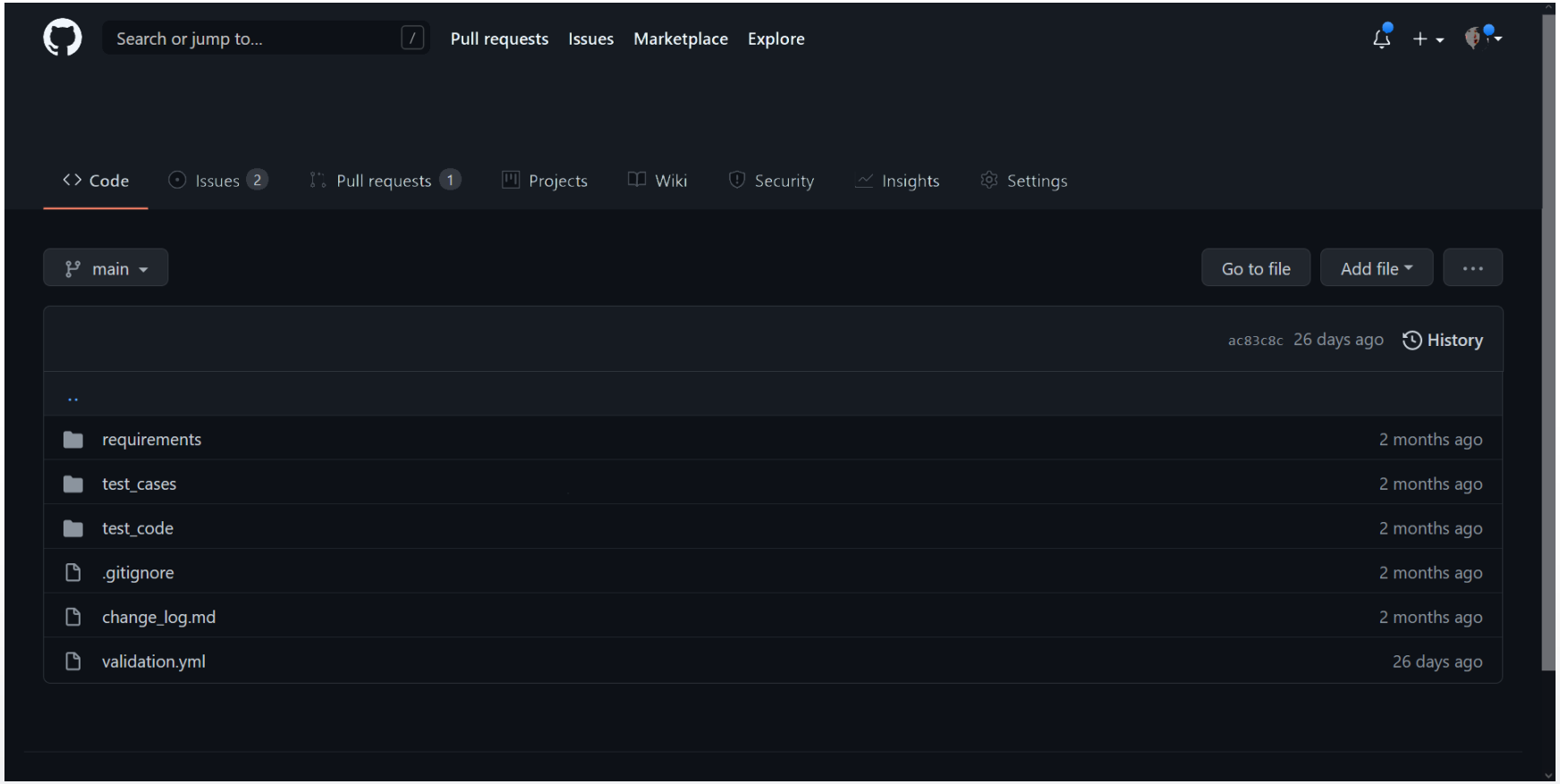
- Read in the file `validation/testdata/rpharma_pp.csv` and save as `this_pp`. Verify that the v "species" in `this_pp` is of class `character`.

Test result

Test	Results	Pass/Fail
TC_A1.1	As expected	Pass

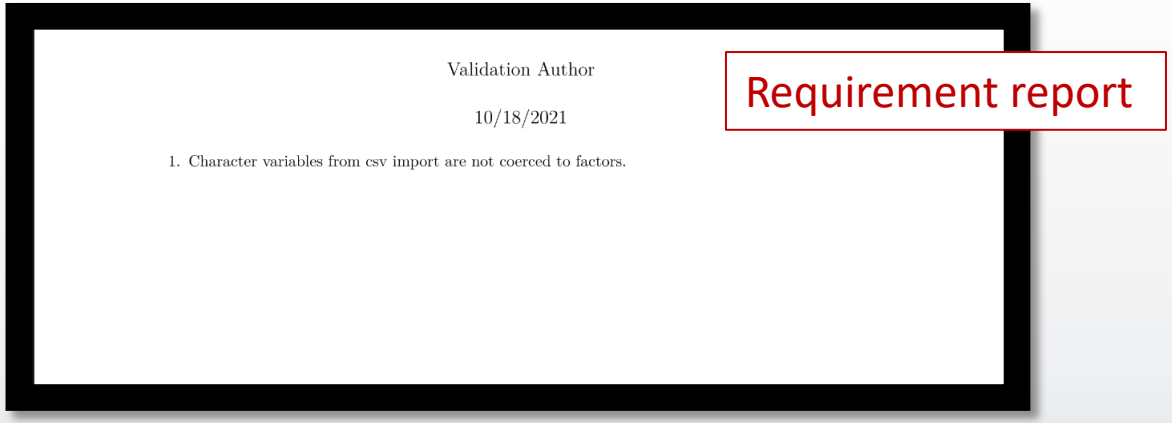
PQ TESTING @ SCHARP

How does validation code evolve over time?



DECISIONS THAT {VALTOOLS} CANNOT HELP WITH …

- What is the default state of the working environment? Is this expected?
- Do we get the same results when switching environments, e.g. different servers or operating systems?
- What custom options are we setting, specific to our organization?
- For workflows that depend on functions from multiple packages, do we observe unexpected interaction?


Validation Author

10/18/2021

1. Character variables from csv import are not coerced to factors.

Requirement report

WHY WE LIKE {VALTOOLS} AT SCHARP

Modular development of requirements, test cases, test code, reports

Easy to customize and reuse validation elements

Extensibility/flexibility as needs change

Parallelization between decision-making and testing phases

Accessible materials for developers and non-developers

Bonus: single paradigm for system validation & package development validation

THANK YOU

Marie Vendettuoli // Senior Statistical Programmer // mvendett@scharp.org