RELM

Testing Procedure and Implementation

Danijel Schorlemmer, Matt Gerstenberger,
Dave Jackson, Stefan Wiemer, Yan Kagan,
Lucy Jones, Ned Field

RELM

Features

3 testing rulesets
quasi-stationary, time-dependent, and real-time models

Parameter uncertainties
error distributions of location, magnitude, and FM angles

Independence probabilities
aftershock vs. main shock

***M_c*-windows**
time and magnitude

Resolution independent
location, magnitude, focal mechanisms angles

Analysis of spatial and magnitude-range performance

RELM

Expectation & Observation

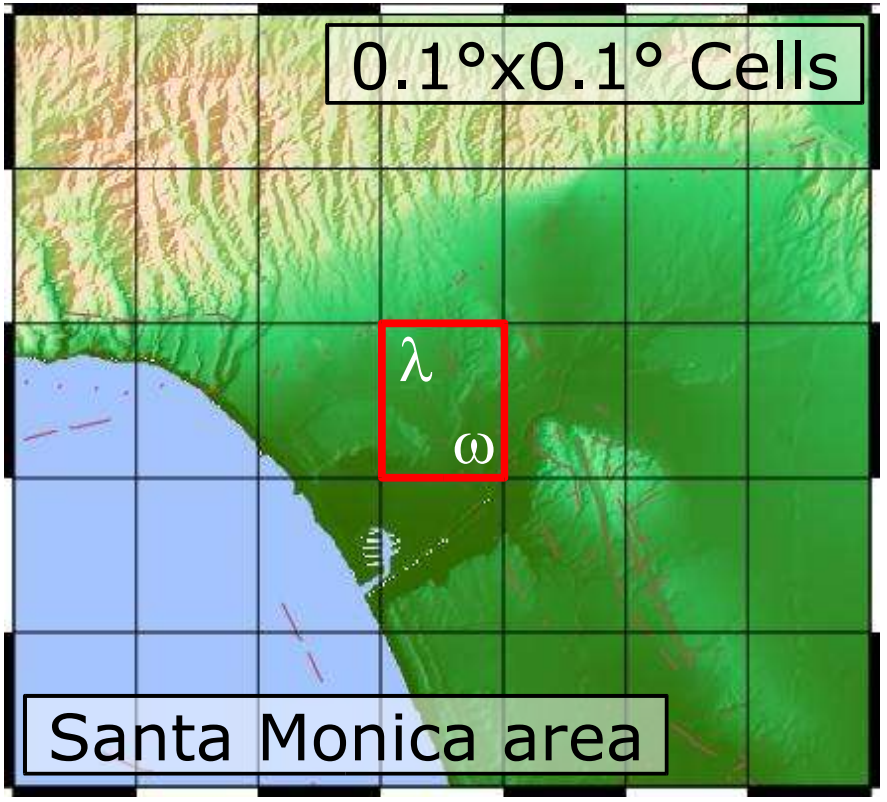
The testing area is separated into cells

A bin defines a volume (cell), magnitude range, and range of focal mechanism angles for which a forecast is issued

In each bin: **Expectation** λ^j
Observation ω

The proposed default binning:

Lon/Lat	0.05°x0.05° (0.1)
Depth	0-30km
Magnitude	0.1
Focal Mech.	30° (none)

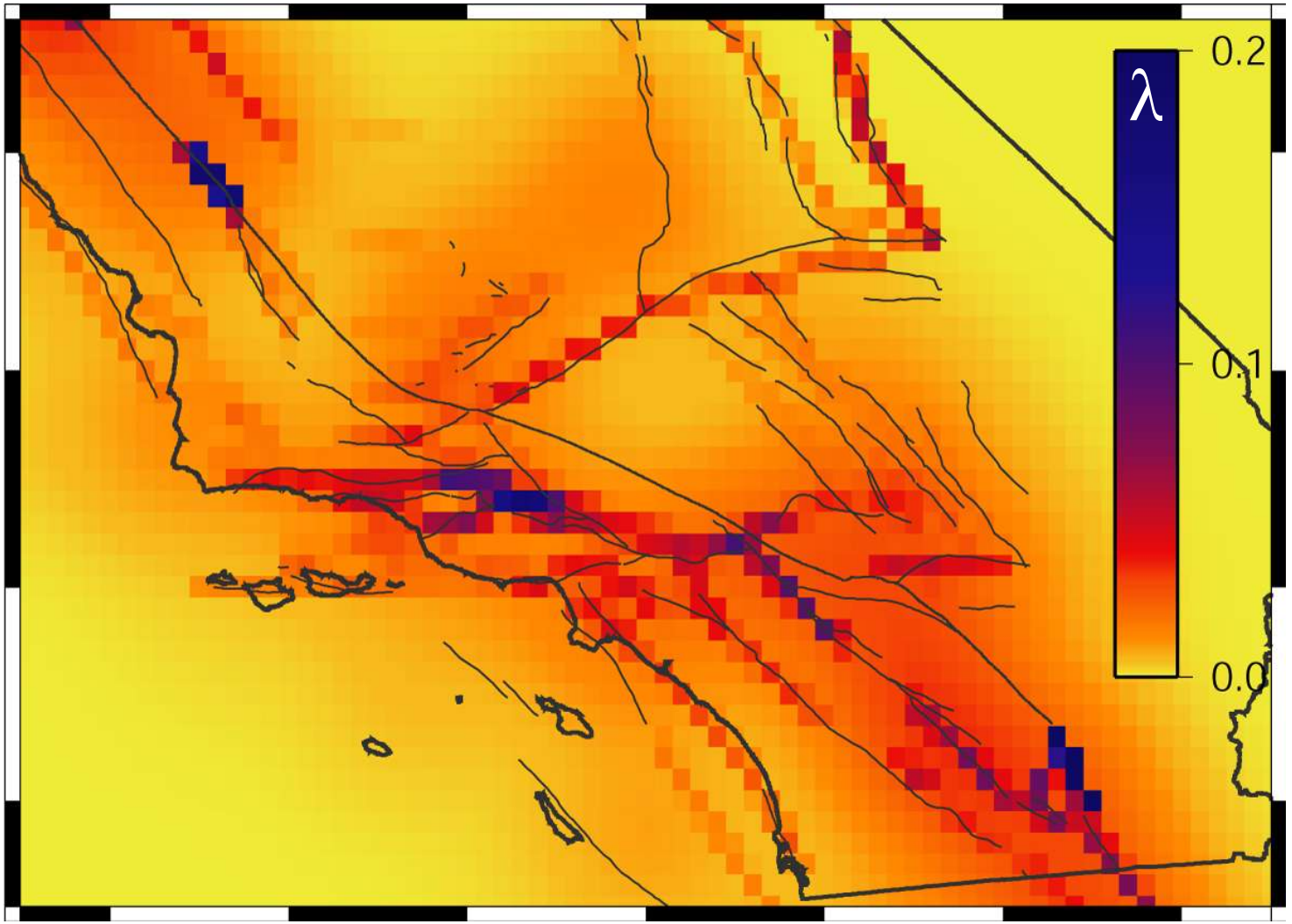


Computing the **likelihood** as the Poissonian probability of making an observation given an expectation.

RELM

Example

Frankel [1996] model (smoothed to grid)



20-year expectations for events of magnitude M ≥ 5

Different Test

We apply 3 different tests:

L-Test
Examines the consistency of a model with the observation (in the likelihood space)

N-Test
Test if the number of observed events is in the range of the expectation of a model

R-Test
Compares 2 models by its log-likelihood-ratio. It estimates the differences in spatial performance.

In each test we compare observed values with the value obtained from simulated catalogs (based on expectations of a model).

RELM

Evaluation

The winning model must:

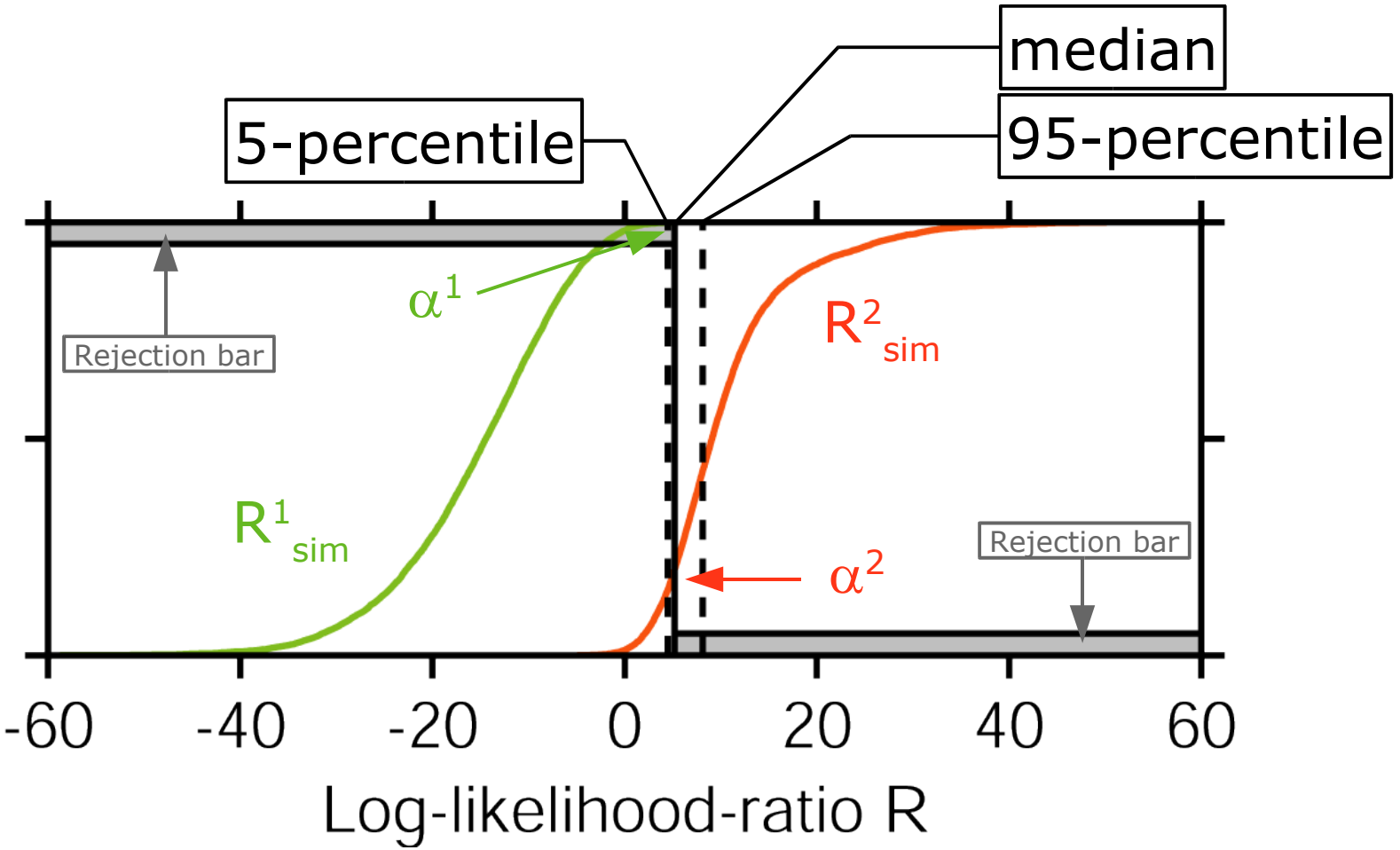
- Beat all other models in the R-Test

- Show consistency with the observation in the L-/N-Test

Visualization

CDF of the values of the simulations

The observed values (including uncertainties) are displayed as vertical lines representing the median and the 5/95 percentiles.



Stationary models

Stationary models require a declustered dataset to test against

Testing schema

Monte Carlo simulations of different declustering algorithms and separate evaluations.

Best possible result: One model wins in all classes
Worst possible result: No decision possible

RELM

Summary

Testing suite for grid-based forecast models

- Rigorous test of fully specified (in advance) forecasts
- 5-years test within the RELM project
- Community-accepted testing algorithms
- Setting a new standard of rigorous forecast tests
- Nucleation point for a RELM-like effort in Europa and worldwide

Testing Center

2 different approaches:

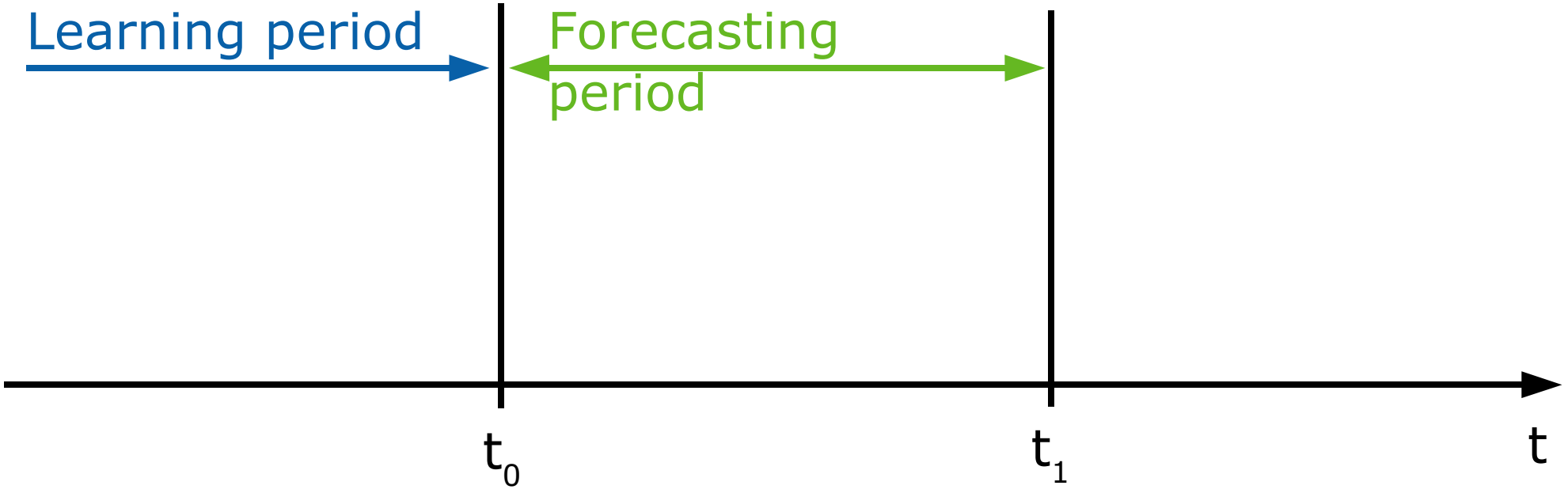
1.) Stationary models deliver the numbers

2.) Time-dependent models (even yearly models) run in the testing center

Why a testing center?

- Re-run the codes with alternative options (different magnitudes)
- Re-run the codes in case of bugs in the testing procedure
- Document each models code and potential changes to it
- Track the modeler's additional data and deposit it
- 'Certify' all steps of testing

Time lines

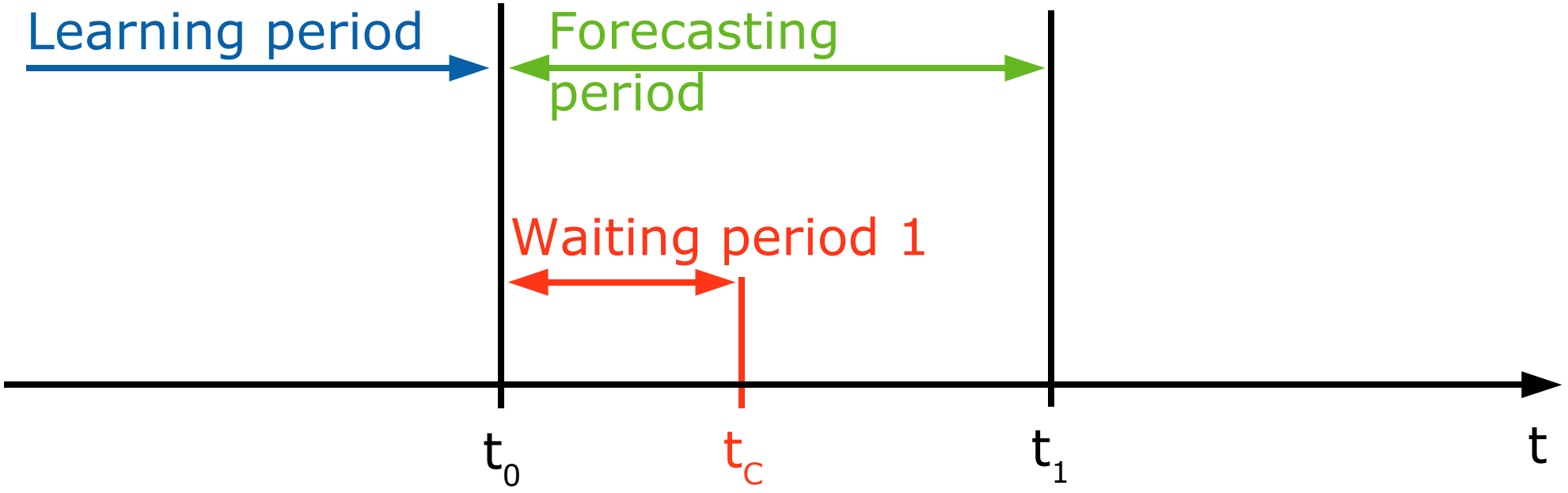


Learning period

Modelers can deposit any data they need for the model, which is not provided by the testing center. This data gets a time-stamp for reproducing results during re-runs.

RELM

Time lines



Waiting period 1

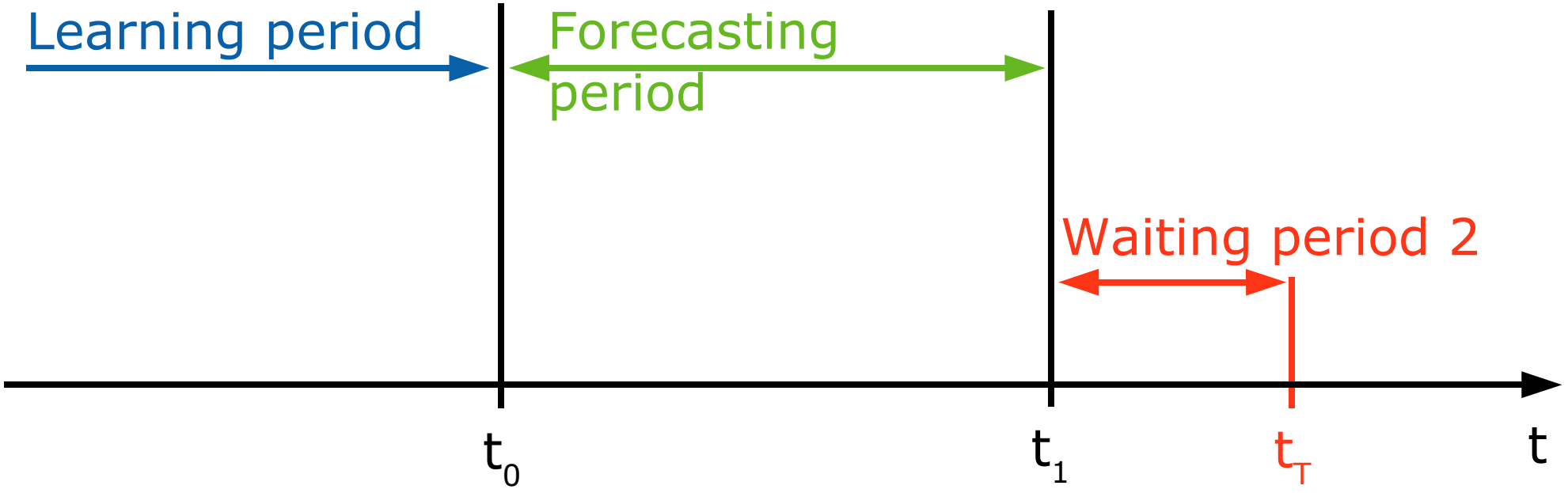
Waiting for the 'authorized' catalog to provide it to models which are seismicity based.

t_c

Computing the forecasts.

RELM

Time lines



Waiting period 2

Waiting for the 'authorized' catalog to test the models against.

t_T

Performing the tests and compute the results.

'Authorized' Data

CISN Data

Our requests:

- Completeness down to magnitude 3.7

- Focal mechanisms (strike/dip/rake of the rupture plane) for all events down to magnitude 4.7

We can include further data sources and 'certify' them.

Testing Center

Each models gets a computer or virtual machine in the testing center:

The modeler installs the operating system and all necessary tools to compile the model's code

The code is deposited in a CVS (Subversion) repository

If the code runs, access to the computer is removed

In case the code crashes, the modeler is allowed to fix the problem: With the CVS, we can track the changes and document them

Web Pages

Display the current results

Interface for retrieving older results

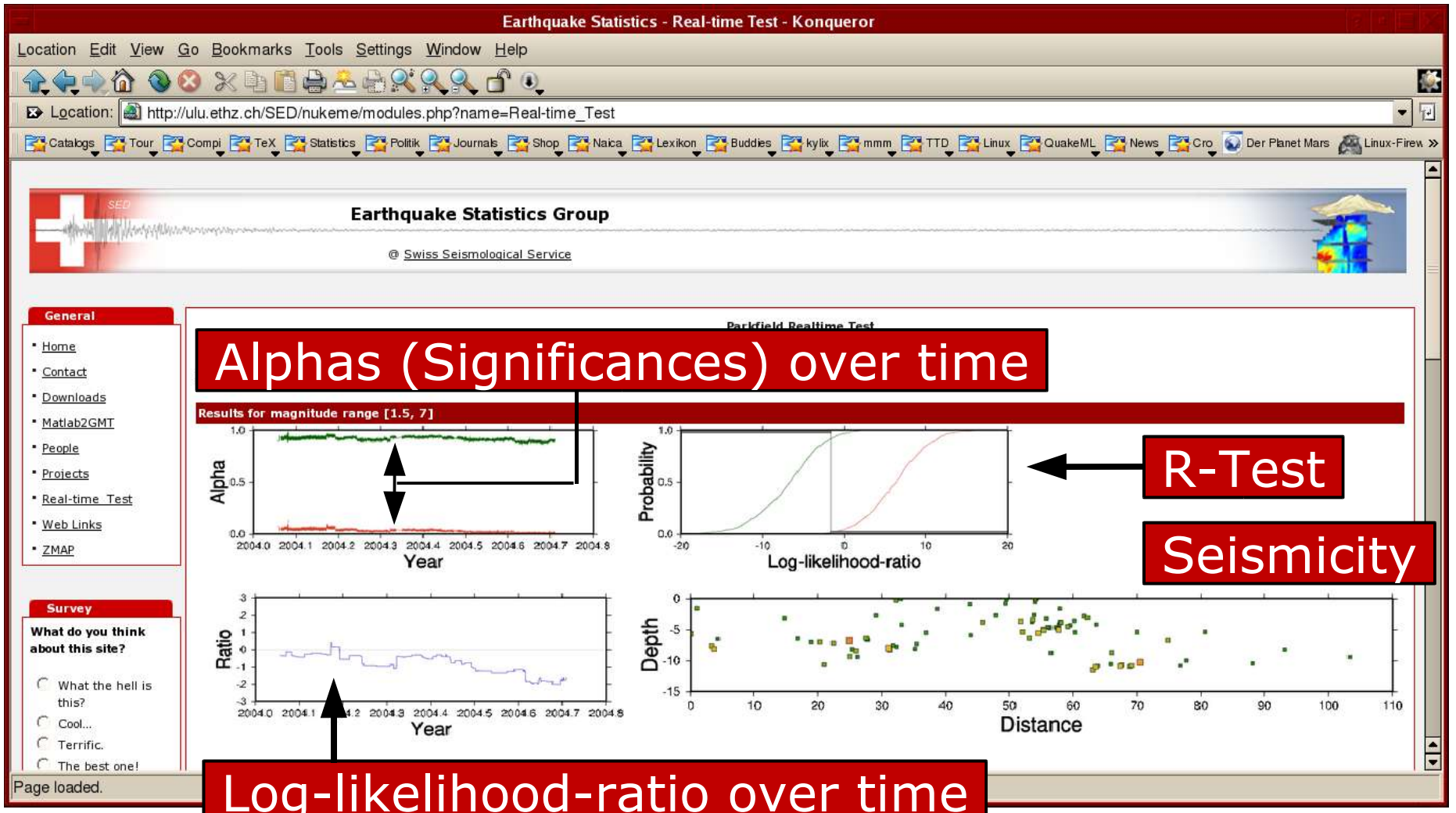
Have all input and output data accessible

Important question

What should be public?

Real-time Test

We implemented a **real-time test** with daily computation of the testresults to track the temporal performance of the 2 models at Parkfield (along a cross-section).



RELM

Next Steps

Investigate the modelers needs

Setting up the testing center

Implement STEP into the testing center as the first model

Subsequently implement the other models