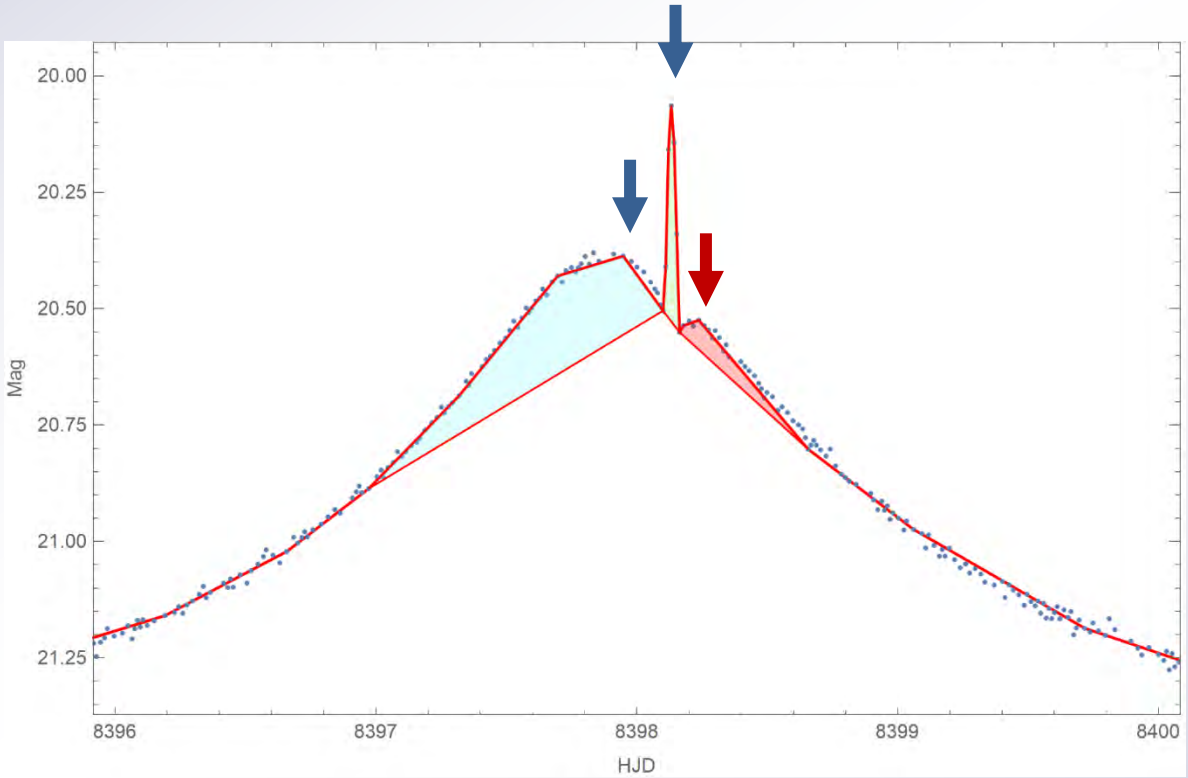
The public release of RTModel:

a platform for the analysis of microlensing events



Valerio Bozza
University of Salerno, Italy

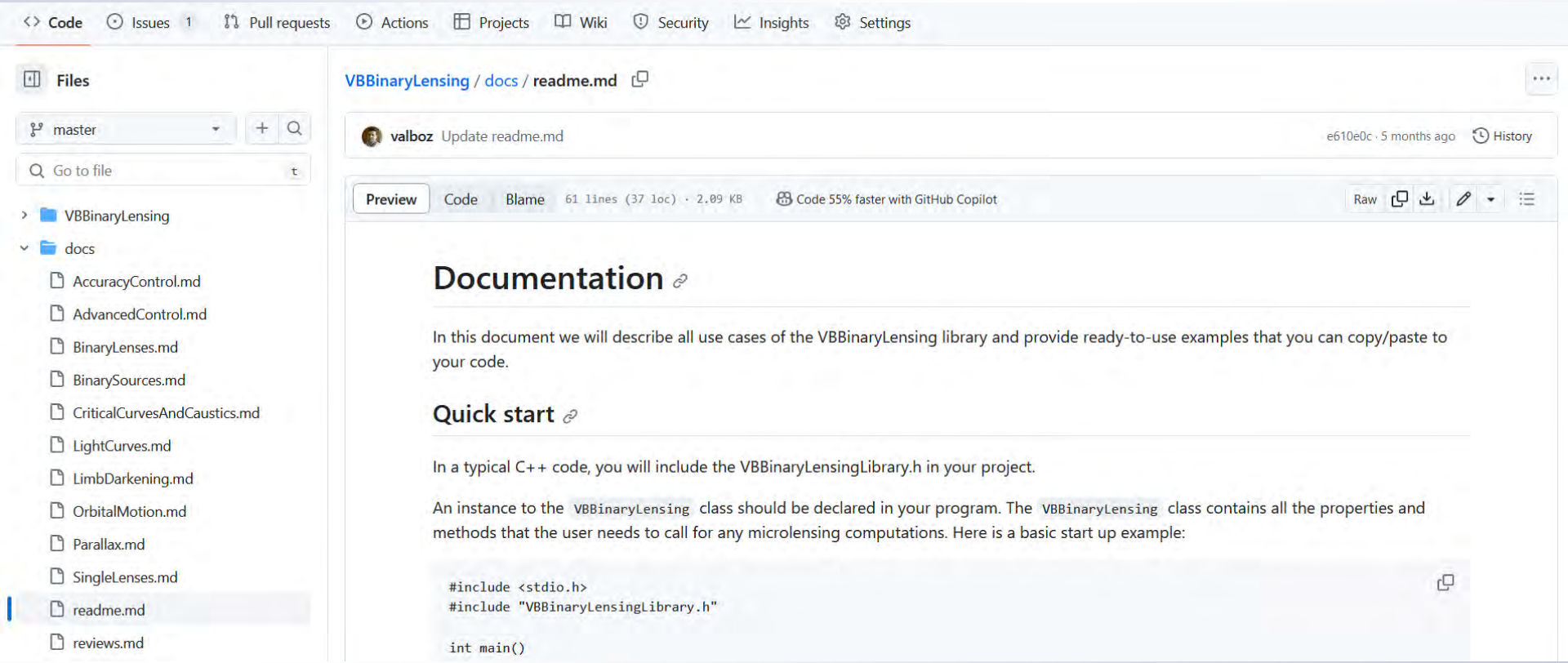
RTModel

- **Real-time microlensing modeling platform**, active since 2011
- Data downloaded from **ARTEMiS** *(Dominik et al.)* via rsync
  http://www.artemis-uk.org/
- Light curves modeled without human intervention in less than 2 hours on a **24-core** (formerly an 8-core) **workstation**.
- Microlensing calculations all performed using **VBBinaryLensing** (published as separate spin-off in 2016).
- Results posted on a **webpage** and publicly available.
  http://www.fisica.unisa.it/GravitationAstrophysics/RTModel.htm
- More than **a thousand events** modeled in real-time.
- Several hundreds modeled off-line.

Year	Events modeled
2011	188
2012	172
2013	129
2014	94
2015	108
2016	130
2017	81
2018	140
2019	65
2020	12
2021	15
2022	12
2023	9
Total	**1155**

VBBinaryLensing

- **VBBinaryLensing** *(VB 2010; VB et al. 2018; VB et al. 2021)*
  - Microlensing computation by contour integration
  - Public code (https://github.com/valboz/VBBinaryLensing)
  - Complete documentation of all functions



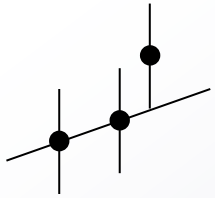
RTModel 2.0

- RTModel is being entirely revised and updated to become a public code *https://github.com/valboz/RTModel*
- Paper explaining all algorithms soon to be submitted.
- **RTModel**: master module (Python) managing the flow and calling all submodules.
- **5 submodules** (C++) interacting via ASCII files.
- Reader: pre-processes the photometry.
- InitCond: sets initial conditions for fitting.
- LevMar: execute the fit for a given model from a given initial condition.
- ModelSelector: select best models for a given category.
- Finalizer: compares models of different categories and assesses event class.

Data pre-processing by Reader

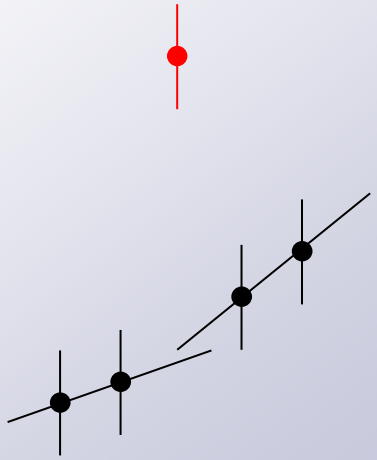
Error bar re-normalization

- Actual scatter of each data set is estimated by summing square residuals from local linear fitting.
- The “chi square” thus obtained is used to re-normalize error bars.



Outliers

- If linear fits from preceding and following points agree but the point is way off these predictions, the point is removed as outlier.



Data pre-processing by Reader

Significance

- A “significance” weight is assigned to each data point based on distance and residuals from neighbors.

Re-binning

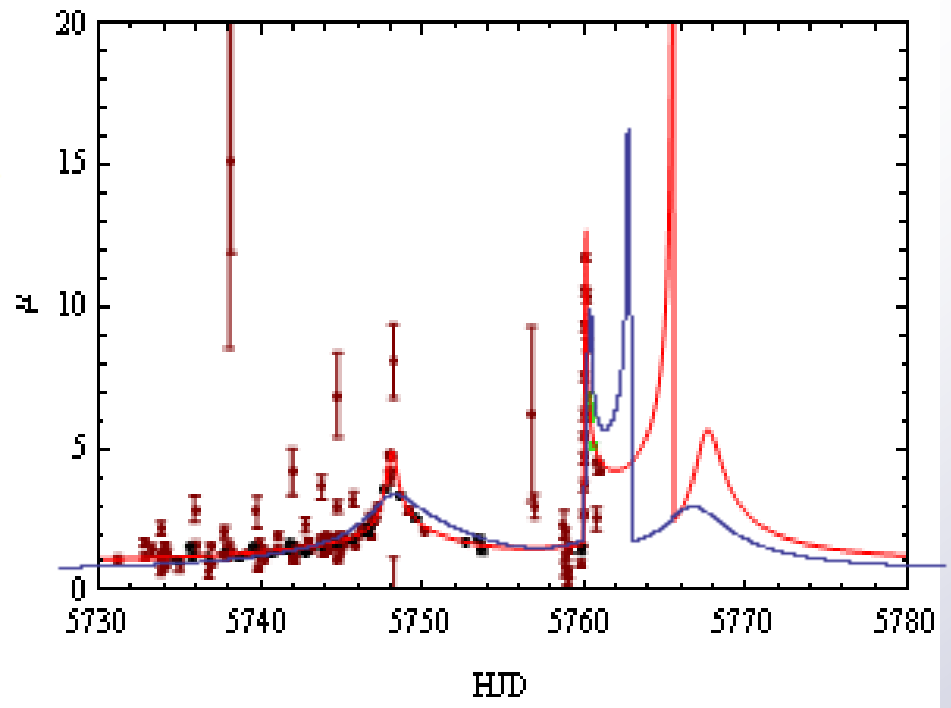
- Low-significance points are replaced by weighted means with their neighbors.

Customization

- User has full control on all pre-processing steps:
  - ✓ renormalization of the error bars
  - ✓ time-correlation threshold
  - ✓ outliers threshold
  - ✓ re-binning level
  - ✓ off-season points usage…

InitCond: template matching

- **Library of 113 template light curves** covering all possible caustic crossings and grazings *(see Mao & Di Stefano 1995; Liebig, D'Ago, VB, Dominik 2015)*.
- For each template we have a **record of peaks** positions.
- By **matching** the peaks in the template to the observed peaks, we fix **(t_0,t_E)**.
- **(s,q,u_0,α,ρ)** are taken from the template.

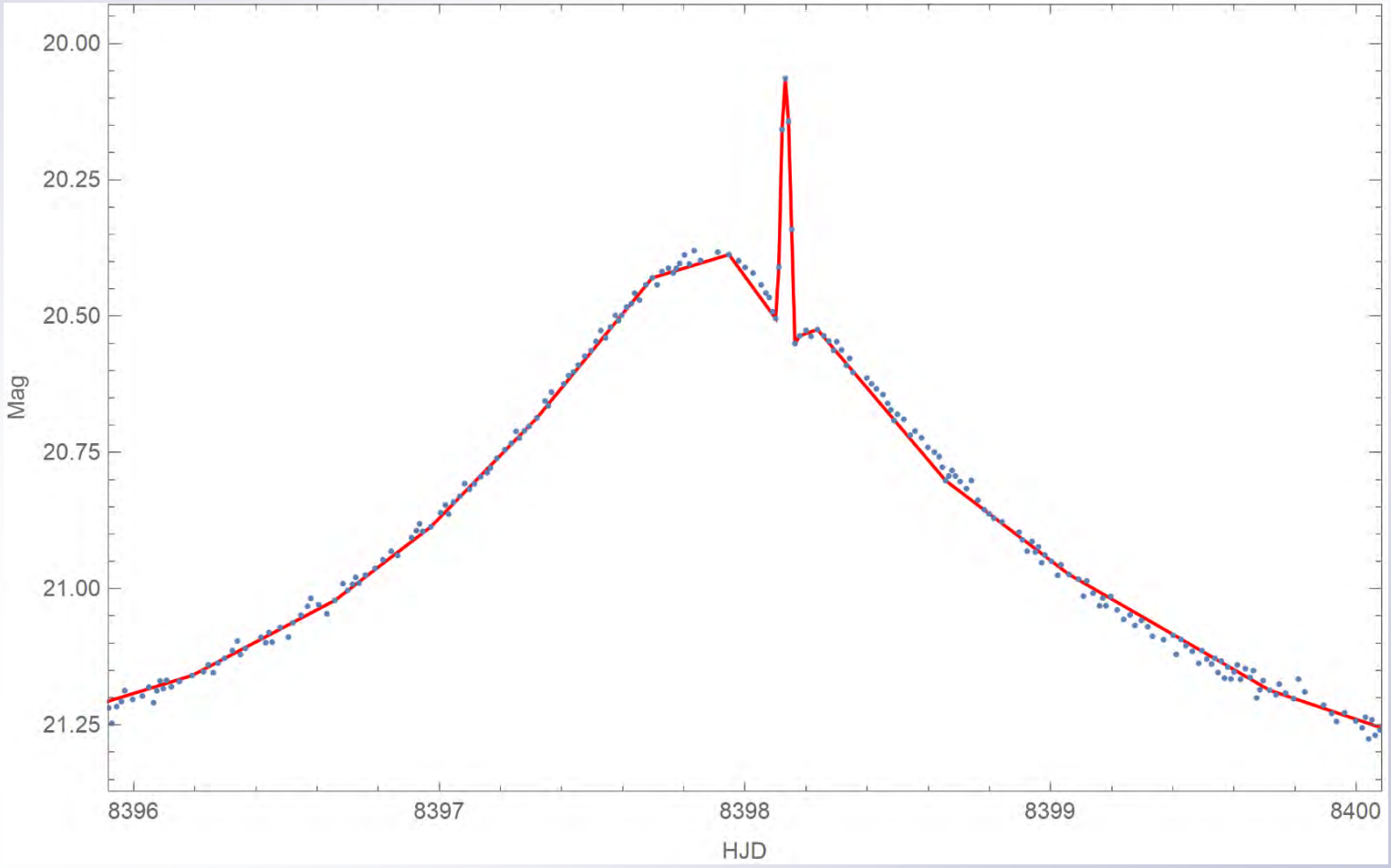


- We run a fit **from each seed** initial condition.

Initial conditions by InitCond

SPLINE approximation

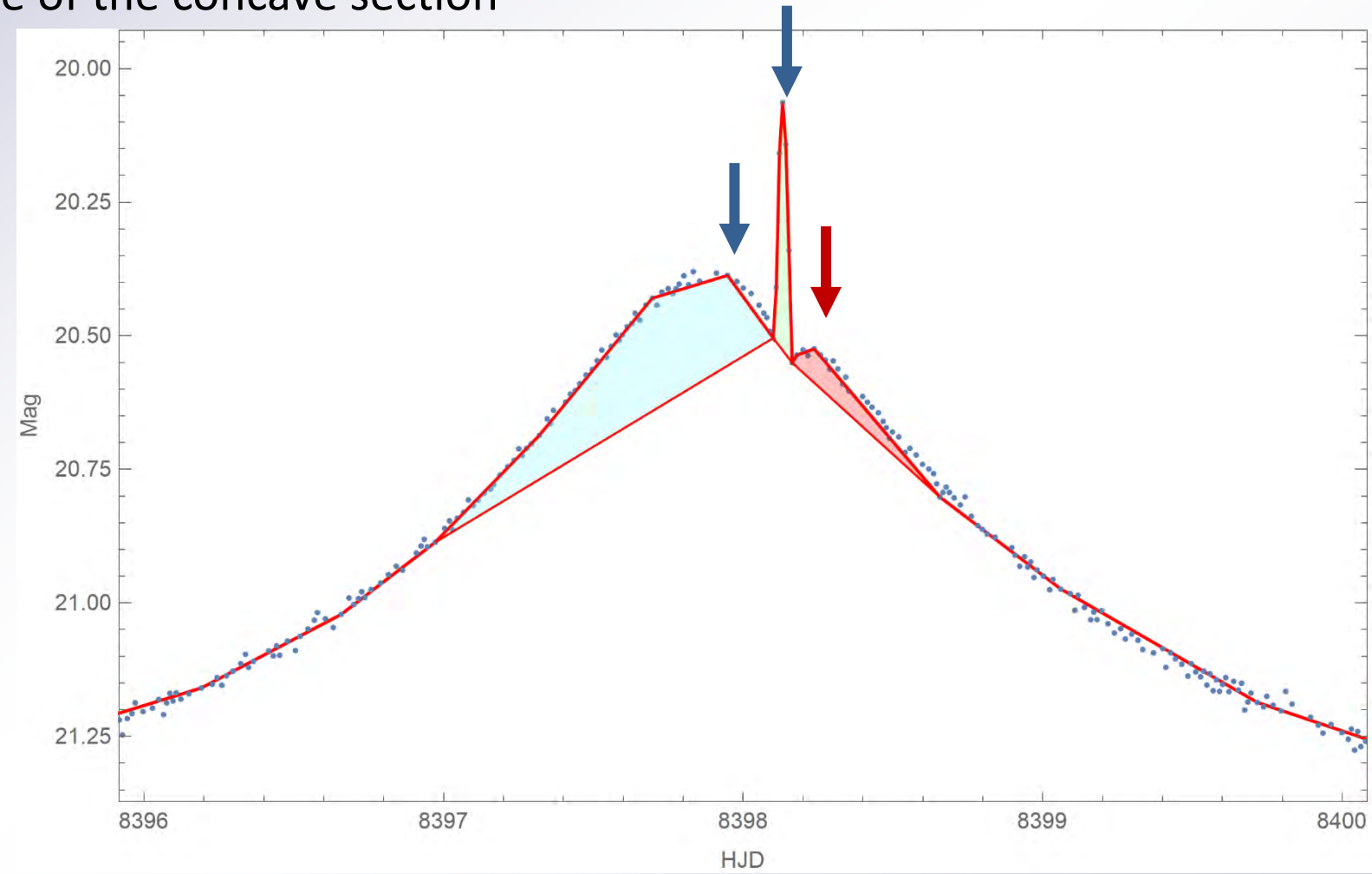
- Each dataset is approximated by a SPLINE until the maximum residual is less than 5σ.



Initial conditions by InitCond

Candidate “peaks” identification

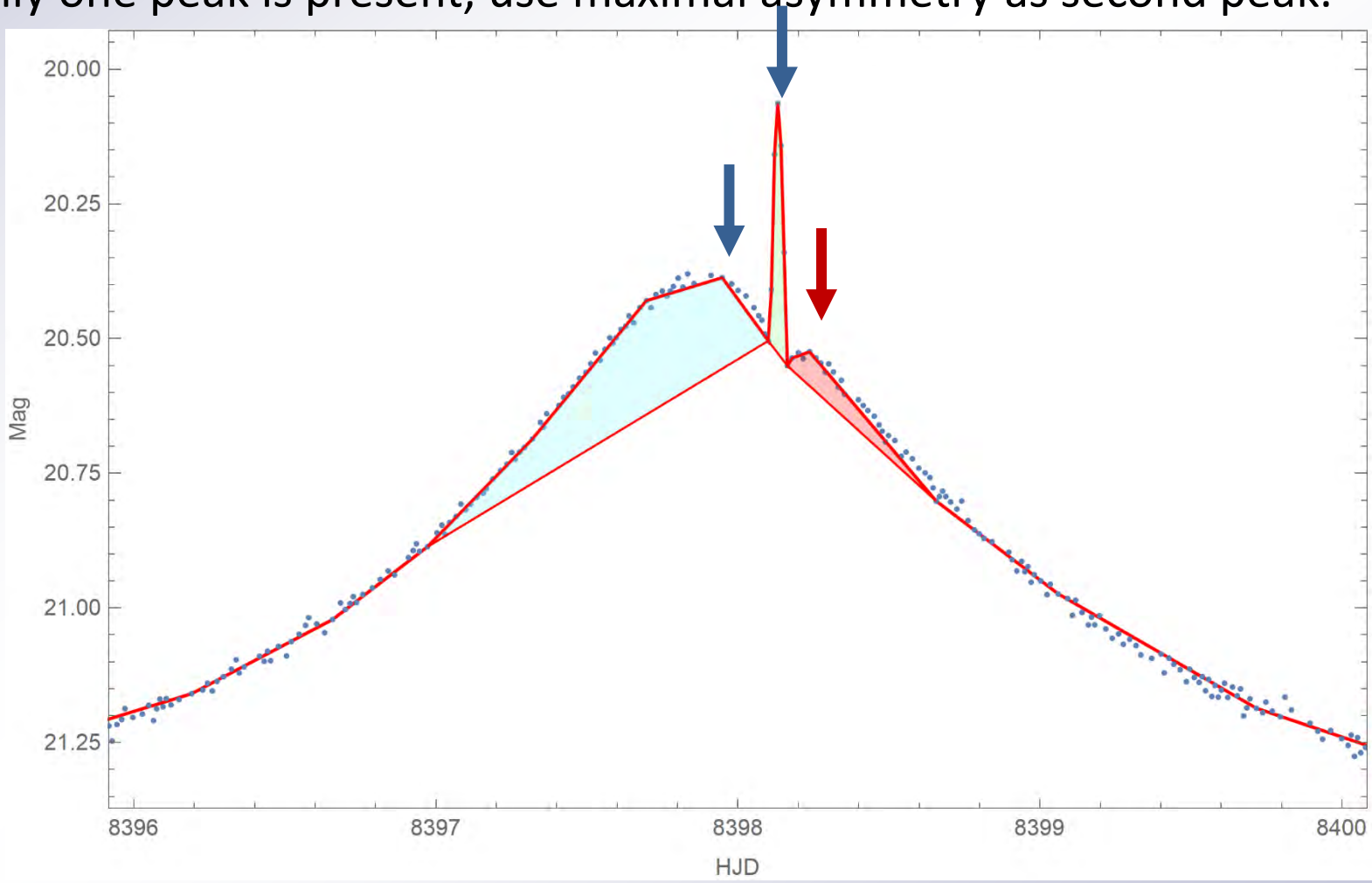
- In each concave section, we find a peak
- If there is no peak, we take the point with the maximal deviation from the base of the concave section



Initial conditions by InitCond

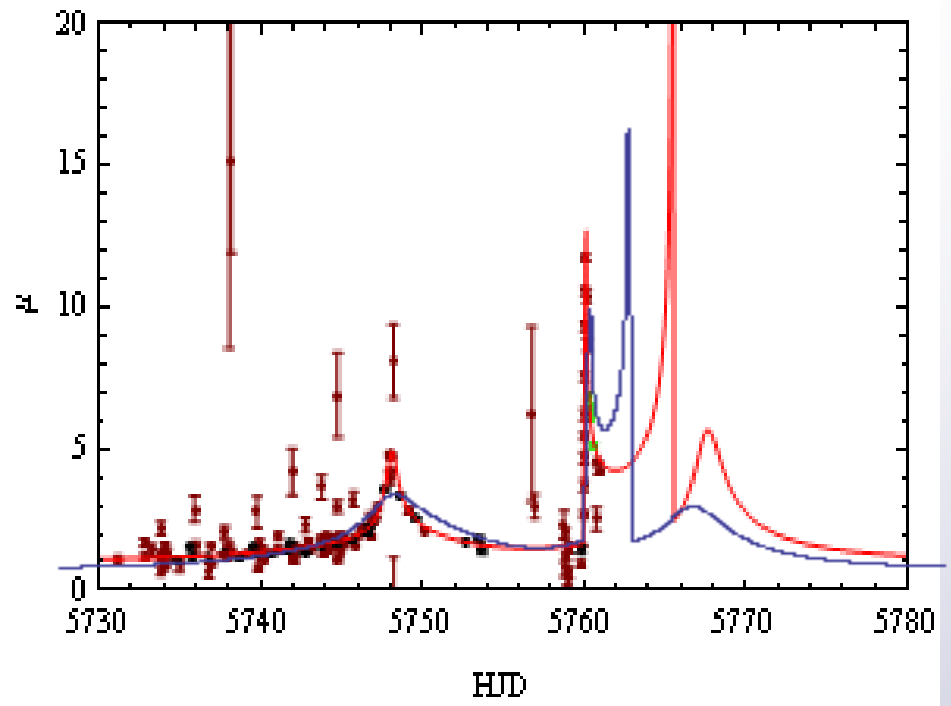
Selection of “peaks”

- Peaks in different datasets are cross-matched. Duplicates are removed.
- A minimum of two most prominent peaks are selected.
- If only one peak is present, use maximal asymmetry as second peak.



InitCond: template matching

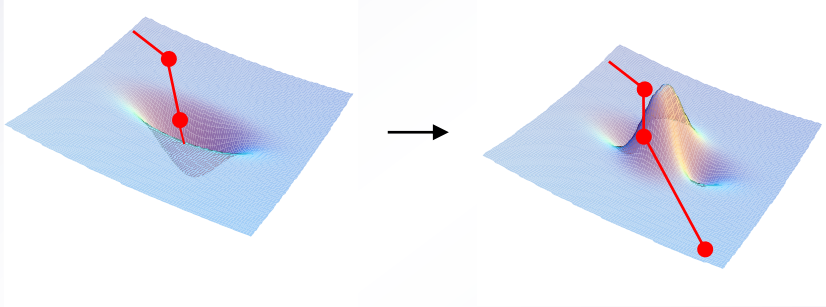
- **Library of 113 template light curves** covering all possible caustic crossings and grazings *(see Mao & Di Stefano 1995; Liebig, D'Ago, VB, Dominik 2015)*.
- For each template we have a **record of peaks** positions.
- By **matching** the template peaks to the observed peaks, we derive **(t_0,t_E)**.
- **(s,q,u_0,α,ρ)** are taken from the template.



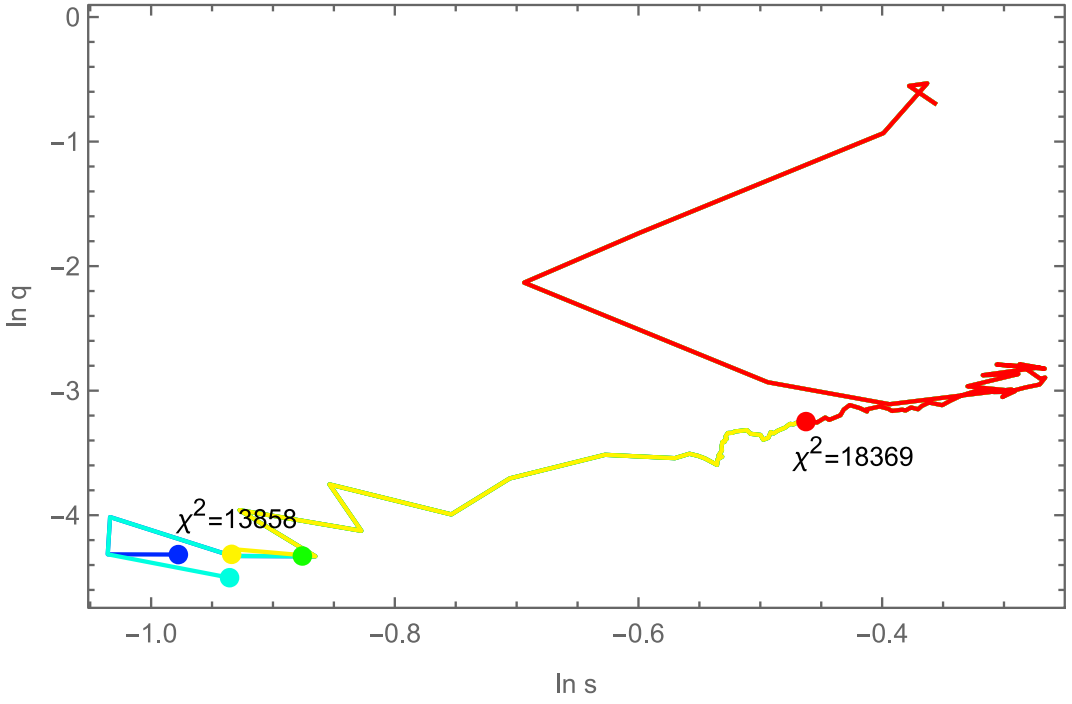
- We run a fit **from each seed** initial condition.

Fitting by LevMar

- **Levenberg-Marquardt** using numerical derivatives.
- Fit repeated several times with each local minimum filled by a **“bumper”** repelling the fit in order to broaden exploration.

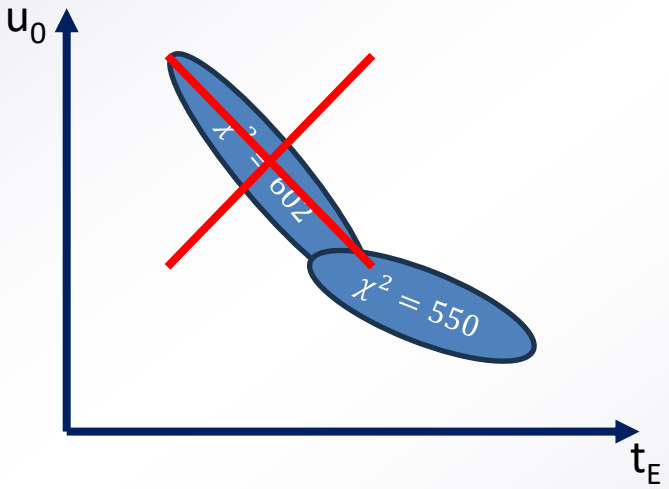


- **Bumpers** are shaped according to the covariance ellipsoid.
- Bumpers are broadened each time the fits fall within them.



ModelSelector: selection within a category

- **Uncertainty** estimated from **Fisher matrix** inversion on each model.
- Models with **uncertainty ellipsoid** overlapping with a better model are discarded.



- Independent models are **retained if** $\Delta\chi^2 < 3\sigma_\chi$, where the standard deviation of the χ^2 distribution is

$$\sigma_\chi = \sqrt{2n} \rightarrow 89 \text{ for n=4000}$$

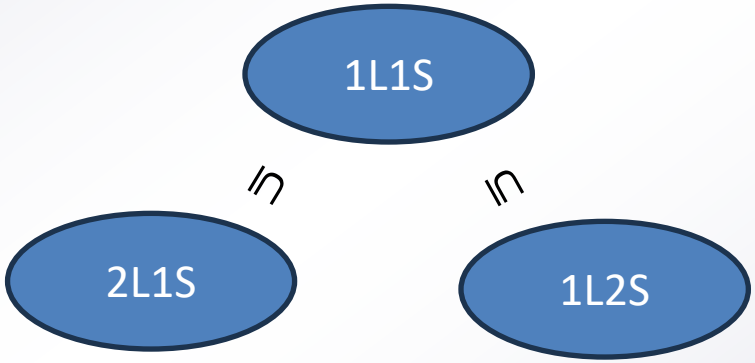
- Note that for all these calculations **error bars are re-normalized** so as to have $\chi^2 = d.o.f. \cong n$ on the best model in the category.

- Models passing this selection are given in output as degenerate models.

Finalizer: Event classification

Nested models

- Nested models are those included in a higher dimensional one
- **Wilks' theorem**: evidence for an alternative model adding *m* parameters vs null hypothesis is tracked by χ^2 distribution with *m* degrees of freedom.
- Only **alternative models improving by** $\Delta\chi^2 > T_m$ with respect to any of their nested models are **retained**.
- The threshold T_m corresponds to a 5σ evidence according to the χ^2 distribution with *m* degrees of freedom



m	T_m	Example
2	40	Parallax models
3	43.5	Circular orbital motion
4	46.5	Binary vs PSPL
5	49.2	Binary w par vs FSPL
6	51.9	Binary w par vs PSPL

Unnested models

- We **retain** unnested models following **the softer criterium** $\Delta\chi^2 < 3\sigma_\chi$.
- If models of different categories pass this criterium, they are given in output as degenerate models.

Conclusions

- RTModel will soon be public:
- 5 C++ modules and a Python master module.
- Broad possibility for customization of all parameters.
- Scriptable for large-scale analysis.
- Fast automatic modeling and classification of microlensing events with no human intervention.
- List of all degenerate models in output with parameters and uncertainties.

Future directions

- Inclusion of astrometry/interferometry.
- Extension to 4 bodies (1 source 3 lenses/ 2 sources 2 lenses)
  (see Vito Saggese's talk)