FERMILAB-FN-0950-APC

Measurements of transverse beam halo diffusion rates in the LHC with collimator scans

G. Stancari,* V. Previtali, and A. Valishev

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510, USA

R. W. Assmann, F. Burkart, S. Redaelli, B. Salvachua, and G. Valentino†

CERN, Geneva, Switzerland

(Dated: September 10, 2012)

Measurements of beam halo dynamics were conducted in the LHC on 22 June 2012 at 4 TeV using collimator scans. Four scans were performed: on proton beam 1 with a vertical collimator and on beam 2 with a horizontal collimator, first with squeezed separated beams and subsequently in collision. The time evolution of losses following collimator steps provides experimental estimates of halo diffusion, halo population, impact parameter distributions, and beam jitter. Besides its general interest for beam dynamics and machine performance characterization, this information forms an essential input for the general design of collimation systems, including new methods such as crystal collimators and hollow electron lenses. In this note, the results of an analysis of the data is presented.

* Corresponding author; E-mail: stancari@fnal.gov

† Experiment coordinator.

Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy.

CONTENTS

I. Introduction 3

II. Model 4

III. Methods and data sets 6

A. Experimental procedure 6

B. Parameters 8

C. Measured quantities 9

IV. Data Analysis 10

V. Results 20

A. Diffusion Coefficients 20

B. Halo Population 22

C. Impact parameter distributions 23

D. Beam Jitter 23

VI. Conclusions 23

VII. Acknowledgments 25

A. Diagnostic plots 26

B. Table of collimator steps 86

C. Table of fit results for each step 93

D. R source code 100

References 152

I. INTRODUCTION

Understanding particle losses and beam quality degradation is one of the fundamental aspects in the design and operation of accelerators. From the point of view of machine protection, losses must be absorbed by the collimation system to avoid damaging components. Beam lifetimes and emittance growth determine the luminosity of colliders. Knowledge of the machine aperture (physical and dynamical) and of the mechanisms that drive particle loss is essential.

The dynamics of particles in an accelerator can be quite complex. Deviation from linear dynamics can be large, especially in the beam halo. Lattice resonances and nonlinearities, coupling, intrabeam and beam-gas scattering, and the beam-beam force in colliders all contribute to the topology of the particles' phase space, which in general will include regular and chaotic regions, and resonant islands. In addition, various noise sources are present in a real machine, such as ground motion (resulting in orbit and tune jitter) and ripple in the radiofrequency and magnet power supplies. As a result, the macroscopic motion can acquire a stochastic character, which can be described in terms of particle diffusion [1–5].

Calculations of lifetimes, emittance growth rates, and dynamic aperture from various sources are routinely performed in the design stage of all major accelerators, providing the foundation for the choice of operational machine parameters. With these studies, we address the corresponding lack of experimental data on the dynamics of the beam halo, with the goal of relating it to these basic parameters.

It was shown that beam halo diffusion can be observed by observing the time evolution of particle losses during a collimator scan [12]. These phenomena were used to estimate the diffusion rate in the beam halo in the SPS at CERN [7], in HERA at DESY [12], and in RHIC at BNL [8]. An extensive experimental campaign was carried out at the Tevatron in 2011 [11] to characterize the beam dynamics of colliding beams and to study the effects of the novel hollow electron beam collimator concept [9]. Following the results of the Tevatron measurements, similar experiments were proposed in the LHC [15].

This report describes the measurement technique and the data analysis in detail. Numerical calculations, data analysis, and graphics were done with the open-source, multi-platform statistical package R version 2.15.1 (2012-06-22) [13]. This documentation was produced by integrating LaTeX with R using the Sweave package [14]. The complete analysis scripts are collected in Appendix D.

II. MODEL

Particle motion at the microscopic level is in general very complex. Two main considerations lead to the *ansatz* that macroscopic motion in a real machine, especially in the halo, will be mostly stochastic: (1) the central limit theorem applied to the multitude of dynamical effects acting on the beam; (2) the operational experience during collimator setup, which generates loss spikes and loss dips which often decay in time as $1/\sqrt{t}$.

A diffusion model of the time evolution of loss rates caused by a step in collimator position was developed [10]. It builds upon the model of Ref. [12] and its assumptions: (1) constant diffusion rate within the range of the step and (2) linear halo tails. These hypotheses allow one to obtain analytical expressions for the solutions of the diffusion equation and for the corresponding loss rates vs. time. The present model addresses some of the limitiations of the previous model and expands it in the following ways: (a) losses before, during, and after the step are predicted; (b) different steady-state rates before and after are explained; (c) determination of the model parameters (diffusion coefficient, tail population, detector calibration, and background rate) is more robust and precise. These calculations are the basis for the measurement of transverse beam diffusion rates as a function of particle amplitude with collimator scans.

Following Ref. [12], we consider the evolution in time t of a beam of particles with phase-space density $f(J,t)$ described by the diffusion equation

$$\partial_t f = \partial_J \left(D \, \partial_J f \right),$$

where J is the Hamiltonian action and D the diffusion coefficient. The particle flux at a given location $J = J'$ is $\phi = -D \cdot [\partial_J f]_{J=J'}$. During a collimator step, the action $J_c = x_c^2/\beta_c$, corresponding to the collimator position x_c at a ring location where the amplitude function is β_c, changes from its initial value J_{ci} to its final value J_{cf} during a time Δt. The step in action is $\Delta J \equiv J_{cf} - J_{ci}$. In the Tevatron, typical steps are 50 μm in 0.2 s, and the amplitude function is tens of meters. It is assumed that the collimator steps are small enough so that the diffusion coefficient can be treated as a constant in that region. This hypothesis is justified by the fact that the fractional change in action is of the order of $\Delta J_c/J_c \sim (2)(25\ \mu\text{m})/(2\ \text{mm}) = 2.5\%$. Because the diffusion coefficient is a strong function of action ($D \sim J^4$), this translates into a variation of 10% in the diffusion rate, an acceptable systematic in a quantity that varies by orders of magnitude. If D is constant, the diffusion equation becomes $\partial_t f = D \, \partial_{JJ} f$. With these definitions, the particle loss rate at the collimator is equal to the flux at that location: $L = -D \cdot [\partial_J f]_{J=Jc}$. Particle showers caused by

the loss of beam are measured with scintillator counters placed close to the collimator jaw. The observed shower rate is parameterized as

$$S = kL + B, \tag{1}$$

where k is a normalization constant including detector acceptance and efficiency and B is a background term which includes, for instance, the effect of residual activation.

Under the hypotheses described above, the diffusion equation can be solved analytically using the method of Green's functions, subject to the boundary condition of vanishing density at the collimator and beyond. Details are given in Ref. [10].

An example of the evolution of the phase-space density according to this model is shown in Figure 1. A few representative snapshots in time are chosen: during collimator movement ($0 \leq t \leq \Delta t$); a short time after the step, with a time scale determined by $\left|J_{ci} - J_{cf}\right|^2 / D = 10$ s; and a long time after the step, with a characteristic time $J_c^2/D = 160$ s.

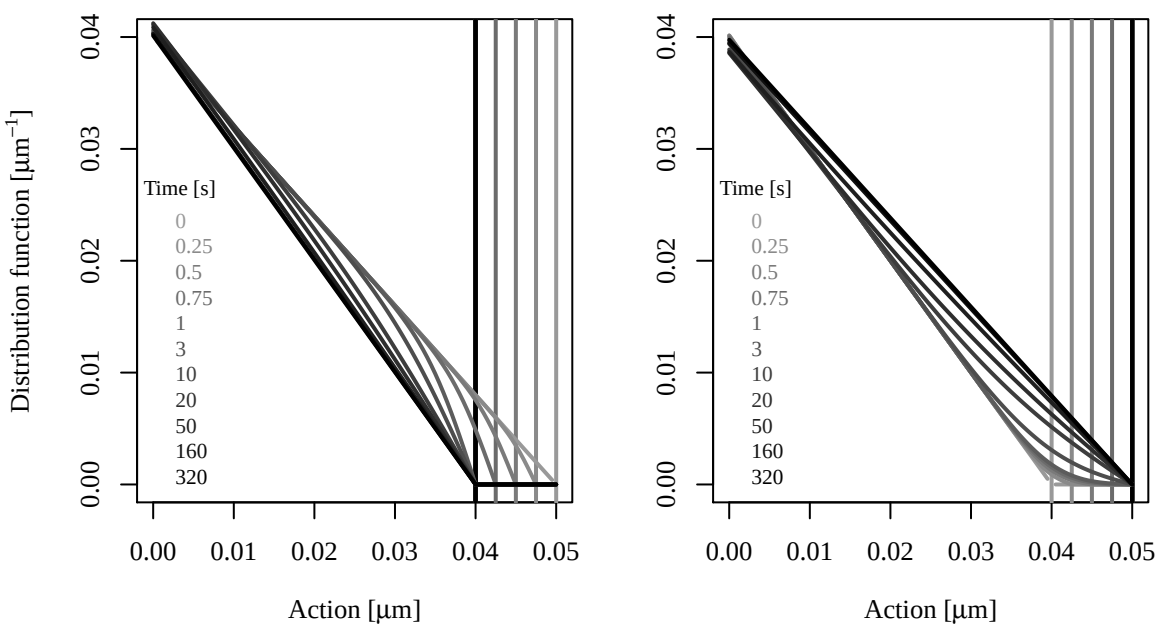


FIG. 1. Evolution of distribution function during collimator step: $f_I(J,t)$ (inward, left) and $f_O(J,t)$ (outward, right). The vertical lines represent the positions of the collimator vs. time. Collimator action varies between $J_{ci} = 0.05$ μm and $J_{cf} = 0.04$ μm in the inward case (viceversa in the outward case) in a time $\Delta t = 1$ s. The slopes of the tails are $A_i = 0.8$ μm^{-2} and $A_f = 1$ μm^{-2} in the inward case (viceversa outwards). The diffusion coefficient is $D = 10^{-5}$ μm^2/s.

Local losses are proportional to the gradient of the distribution function at the collimator. The gradients differ in the two cases of inward and outward step, denoted by the I and O subscripts,

respectively:

$$\partial_J f_I(J_c,t) = -A_i + 2(A_i - A_c)P\left(\frac{-J_c}{\sigma}\right) + \frac{1}{\sqrt{2\pi}\sigma}\cdot$$
$$\cdot\left\{-2A_i(J_{ci}-J_c) + 2(A_iJ_{ci} - A_cJ_c)\exp\left[-\frac{1}{2}\left(\frac{J_c}{\sigma}\right)^2\right]\right\} \tag{2}$$

$$\partial_J f_O(J_c,t) = -2A_iP\left(\frac{J_{ci}-J_c}{\sigma}\right) + 2(A_i - A_c)P\left(\frac{-J_c}{\sigma}\right) +$$
$$+2\frac{A_iJ_{ci} - A_cJ_c}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{J_c}{\sigma}\right)^2\right]. \tag{3}$$

The parameters A_i and A_f are the slopes of the distribution function before and after the step, whereas A_c varies linearly between A_i and A_f as the collimator moves. The parameter σ is defined as $\sigma \equiv \sqrt{2Dt}$; its effect is to expose the dependence of losses on the inverse square root of time, as is typical for diffusion processes. The function $P(x)$ is the S-shaped cumulative Gaussian distribution function: $P(-\infty) = 0$, $P(0) = 1/2$, and $P(\infty) = 1$.

The above expressions are used to model the measured shower rates. A few examples are plotted in Figure 2.

Parameters are estimated from a least-squares fit to the experimental data. The background B is measured before and after the scan when the jaws are retracted. The calibration of kA_i and kA_f is estimated by comparing the level of losses with beam intensity and lifetime. In practice, D is determined by both the measured relaxation time and by the value of the peak (or dip) in losses.

The model explains the data very well when the diffusion time is long compared to the duration of the step. With this technique, the diffusion rate can be measured over a wide range of amplitudes. At large amplitudes, the method is limited by the vanishing beam population and by the fast diffusion times. The limit at small amplitudes is given by the level of tolerable loss spikes.

III. METHODS AND DATA SETS

A. Experimental procedure

The experimental procedure is schematically shown in Figure 3. The LHC experiments (MD of 22 June 2012) were described in Ref. [16]. The experimental procedure is reproduced here for convenience. One nominal bunch (1.15×10^{11} protons) per beam was used. The study started with squeezed, separated beams at 4 TeV. Orbit stabilization was turned off. The IR7 primary and secondary collimators were retracted from their nominal settings of 4.3σ and 6.3σ respectively to

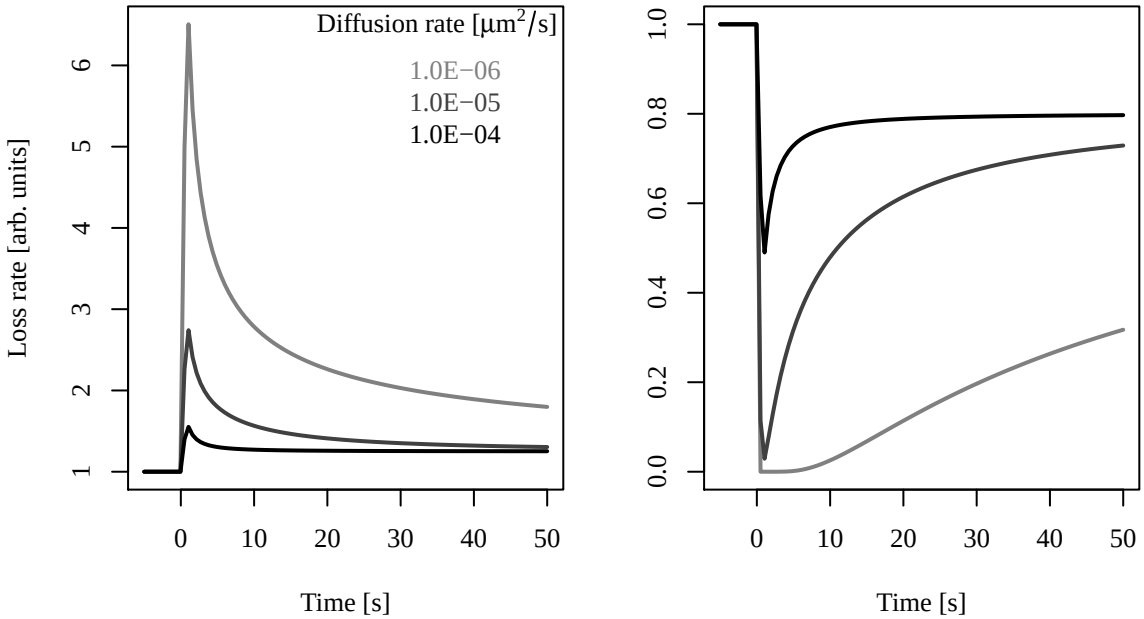


FIG. 2. Calculated evolution of loss rates $L(t)$ during a collimator step according to Eqs. 2 and 3: inward (left) and outward (right). Collimator action varies between $J_{ci} = 0.05$ μm and $J_{cf} = 0.04$ μm in the inward case (viceversa in the outward case) in a time $\Delta t = 1$ s (see also Figure 1). The effect of 3 different values of the diffusion coefficient D is shown. The slopes of the tails are scaled so that the initial and final steady-state loss rates are the same in all cases: $A_i = 1/D$, $A_f = A_i J_{ci}/J_{cf}$.

a half gap of 7σ. The left jaw of the TCP.D6L7.B1 (beam 1, vertical plane) and the TCP.C6R7.B2 (beam 2, horizontal plane) primary collimators were moved in small steps. The collimators were selected from different beams to be able to perform the scrapings in parallel without inducing cross-talk in the BLM signals. The jaws were moved after waiting for the beam losses from the previous step to reach a steady-state (approximately every 10 to 40 seconds). The jaws were left for a few minutes in the beam after they had reached their final inward position, to allow the losses to stabilize. Subsequently, the jaws were moved out in small steps, again after waiting for the transient to decay. The procedure of inward and outward steps was then repeated after bringing the beams into collision.

The electronic logbook is accessible from outside CERN through the following link: https://ab-dep-op-elogbook.web.cern.ch/ab-dep-op-elogbook/elogbook/secure/eLogbook.php, by selecting the LHC OP e-log and the date of the shift.

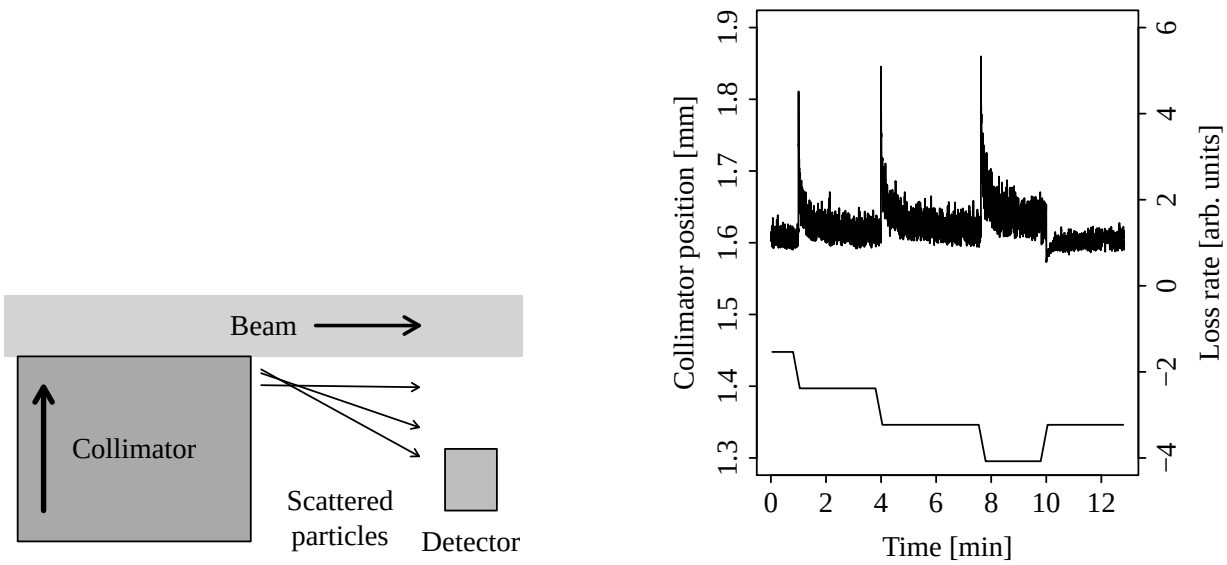


FIG. 3. Schematic diagram of the apparatus (left). Example of the response of local loss rates to inward and outward collimator steps (right).

B. Parameters

The constant parameters used in this analysis are defined here: kinetic energy 4 TeV; relativistic factor $\gamma_r = 4264$; nominal emittances 3.75 μm (normalized), $8.79e-10$ m (geometrical); momentum spread 1.10e-04.

	s [m]	β_x [m]	β_y [m]	D_x [m]	D_y [m]	σ_x^{nom} [mm]	σ_y^{nom} [mm]
TCP.B1.h	19791.2	150.5	82.8	0.33	0.00	0.366	0.270
TCP.B1.v	19789.2	158.8	78.3	0.34	0.00	0.376	0.262
TCP.B2.h	6461.7	150.4	82.8	−0.14	0.00	0.364	0.270
TCP.B2.v	6459.7	158.7	78.3	−0.15	0.00	0.374	0.262

TABLE I. Lattice functions and nominal beam sizes at the collimators of interest.

The lattice functions at the collimators are reported in Table I, together with the nominal beam sizes.

The beam centers from the March 2012 collimator alignment campaign are 0.24 mm (B1, vertical) and 0.52 mm (B2, horizontal). The calibration constant of losses at the TCP.C collimators is approximately $1.250e+12$ p/Gy.

C. Measured quantities

The halo diffusion measurements rely on beam intensities, emittances, bunch length and momentum spread. Moreover, the collimator positions and local losses need to be accurately recorded. Here is a detailed description of the variables used. In particular, these measurements were made possible by the BLM data logged at 12.5 Hz. From inside CERN, the datalogger is accessible at the following link: http://www.cern.ch/timber.

Intensity: The beam current transformer data (BCT) is normally logged every 1 s. For this particular study it was also logged every 20 ms (as suggested by Massimo Giovannozzi, using Riccardo De Maria's dedicated acquisition program). The 20-ms data was missing for about 10 minutes (due to a data acquisition program crash). The format of the BCT data is usually in 2 columns: Unix time stamp (milliseconds since 1970-01-01 00:00 UTC for the 1-Hz data, nanoseconds for the 20-ms data) and number of particles. The 20-ms data tables show some repeated values.

Wire-scan emittances: Three sets of wire scans (WS) were taken: 1) squeezed beams, before scrape; 2) squeezed beams, after scrape; 3) colliding beams before scrape. Some measurements may not be accurate due to saturation (peak values above 8×10^3 arb. units).

Synchrotron-light emittances: Synchrotron-light emittance measurements (BSRT) were logged continuously every 3 s. The raw data files report the Unix time, the measured distribution r.m.s., and the calculated emittance.

Bunch length: Profiles from the longitudinal density monitor (LDM) for both beams were taken at flat top and after the scrape with squeezed beams. Data files include sampled data for a full turn (89 μs) every 25 ps. Bunch length r.m.s. is about 0.3 ns. Satellite bunches are present but likely negligible for our purposes. Estimates of the momentum spread can be obtained from the longitudinal profile and from the parameters of the accelerating cavities.

Momentum spread: Measured in SPS?

Collimator positions: All collimators were moved out after the ramp, starting around 7 times the nominal σ. Collimator scans were done with TCP.D6L7.B1 (vertical, beam 1) and TCP.C6L7.B2 (horizontal, beam 2). Positions were logged every second. The variable labeled MOTOR is the set point, LVDT is the readback. LU is left jaw, upstream; analogously for LD, RU, and RD. Upstream and downstream ends were moved in parallel.

Losses: The beam loss monitor (BLM) data was used. The typical background is 2×10^{-7} Gy/s, noise is 10^{-8} Gy/s, and the signal varies from 10^{-7} Gy/s to about 10^{-3} Gy/s (close to the beam abort threshold). The BLM description file contains BLM name, position along the ring, and a flag (0 = cold magnet, 1 = warm magnet, 3 = collimator). BLMs were logged both at 1 Hz and at 12.5 Hz. The 1-Hz data has an integration time of 1.3 s. Data files were saved every 10 min. They only contain ionization-chamber data. The 12.5-Hz BLM data has an integration time of 82 ms. Fast BLM files have columns labeled by s-position. All detectors are included. If there are two columns with the same position, the first one is the ionization chamber (BLMEI) and the second one is the SEM (BLMES, not used). All measurements are in Gy/s.

IV. DATA ANALYSIS

The evolution of the beam intensity and of the collimator positions during the study is shown in Figure 4. (There is a discrepancy between the 1-s and the 20-ms intensity data. Moreover, the 1-s data shows abrupt spurious dips which are not correlated to losses. The 20-ms data will be used in the analysis.)

An overview of the collimator scans is shown in Figures 5 (separated beams) and 6 (colliding beams). Local losses are defined as the sum of the BLMs closest to the primary collimator of interest: BLMEI.06L7.B1E10 TCP.[A,B,C,D]6L7.B1 (table columns 2811, 2809, 2807, and 2805 in Gianluca's files) for TCP.D6L7.B1 and BLMEI.06R7.B2I10 TCP.[A,B,C,D]6R7.B1 (table columns 2970, 2972, 2974, and 2976) for TCP.C6R7.B2. A comparison of the responses of the 4 BLMs closest to the collimator of interest is shown in Figure 7. The closest BLM is not the most sensitive (D for vertical, C for horizontal); the most sensitive is usually A (4–6 m downstream of the jaw), probably because of the forward evolution of the particle shower.

The background losses are determined during "quiet time," with no beam in the machine (between 2012-06-22 05:28:00 and 2012-06-22 05:48:00). For the local B1 BLMs the background level is $1.81e-06$ Gy/s, and $1.06e-06$ Gy/s for beam 2. For total losses, it is 0.929 Gy/s.

For calibrating the response of this group of BLMs (conversion from Gy/s to protons/s), the experiment time is subdivided into 20-s intervals to obtain the average collimator position, local loss rate, and intensity decay rate. The BLM loss rate vs. total particle loss rate is shown in Figure 8. As expected, there is a systematic shift between separated and colliding beams in the total loss rate (all BLMs around the ring), due to luminosity. The local losses (4 BLMs as defined

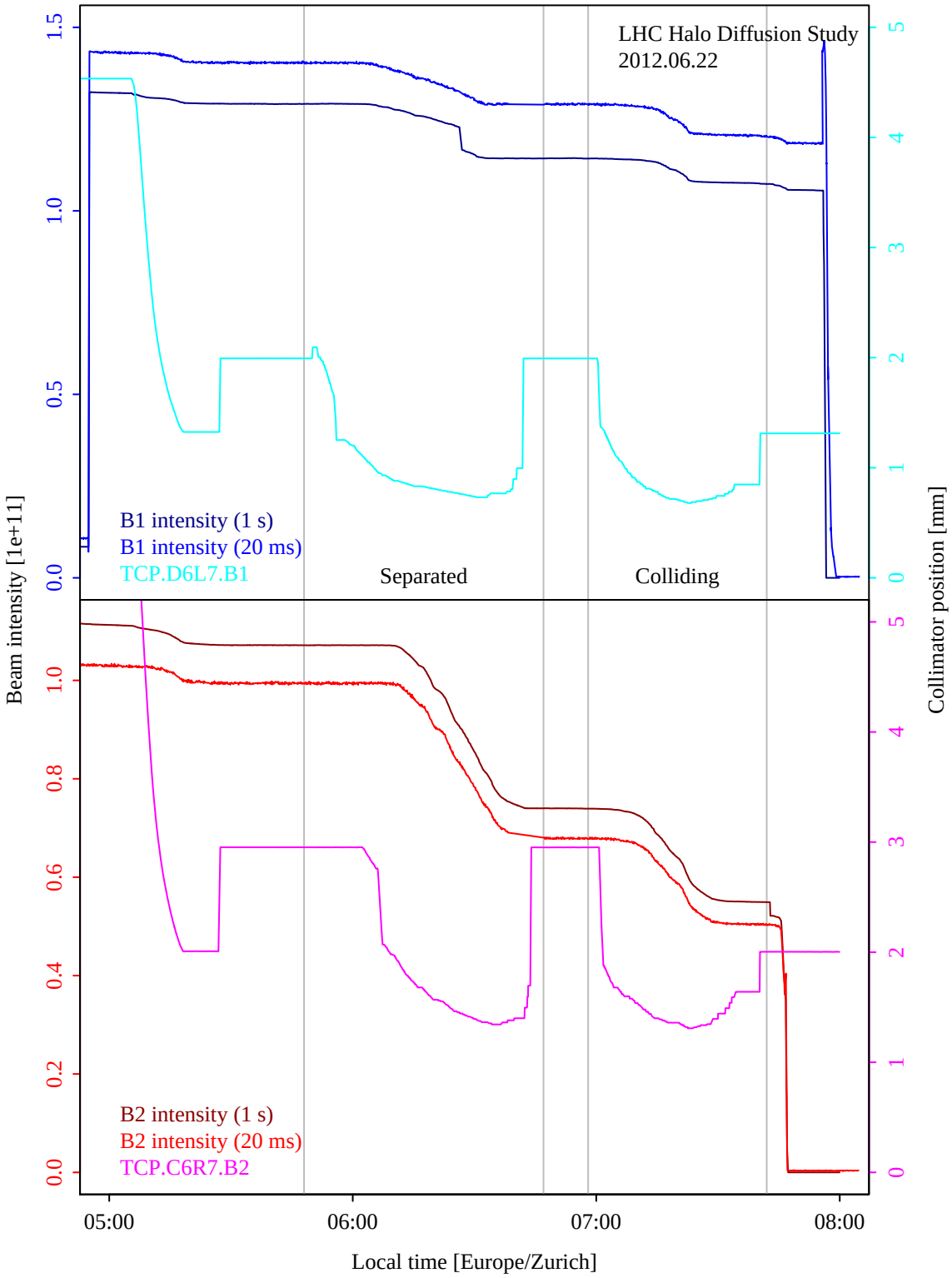


FIG. 4. Beam intensities and collimator positions over the course of the study.

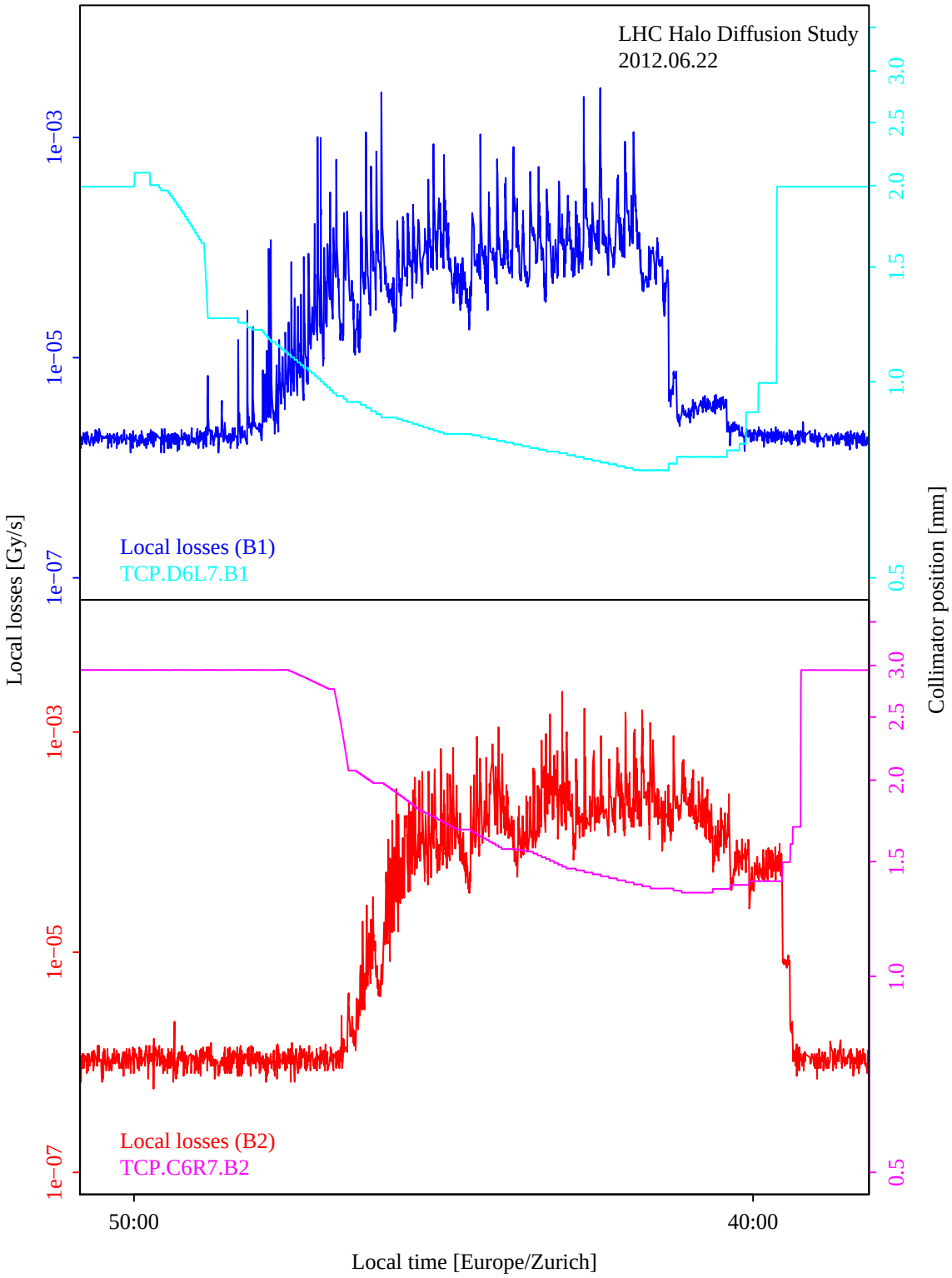


FIG. 5. Local losses and collimator positions during the collimator scan with separated beams.

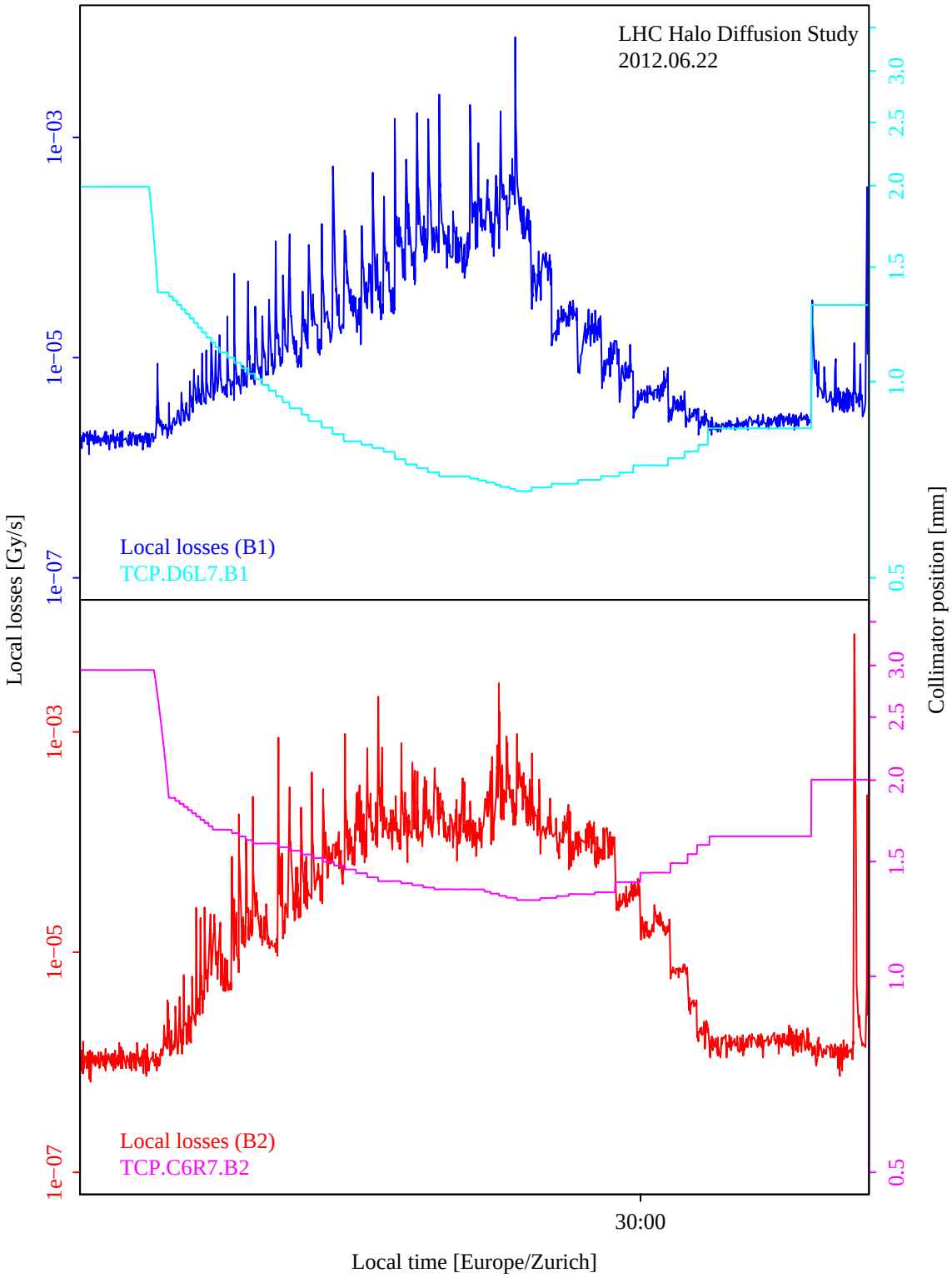


FIG. 6. Local losses and collimator positions during the collimator scan with colliding beams.

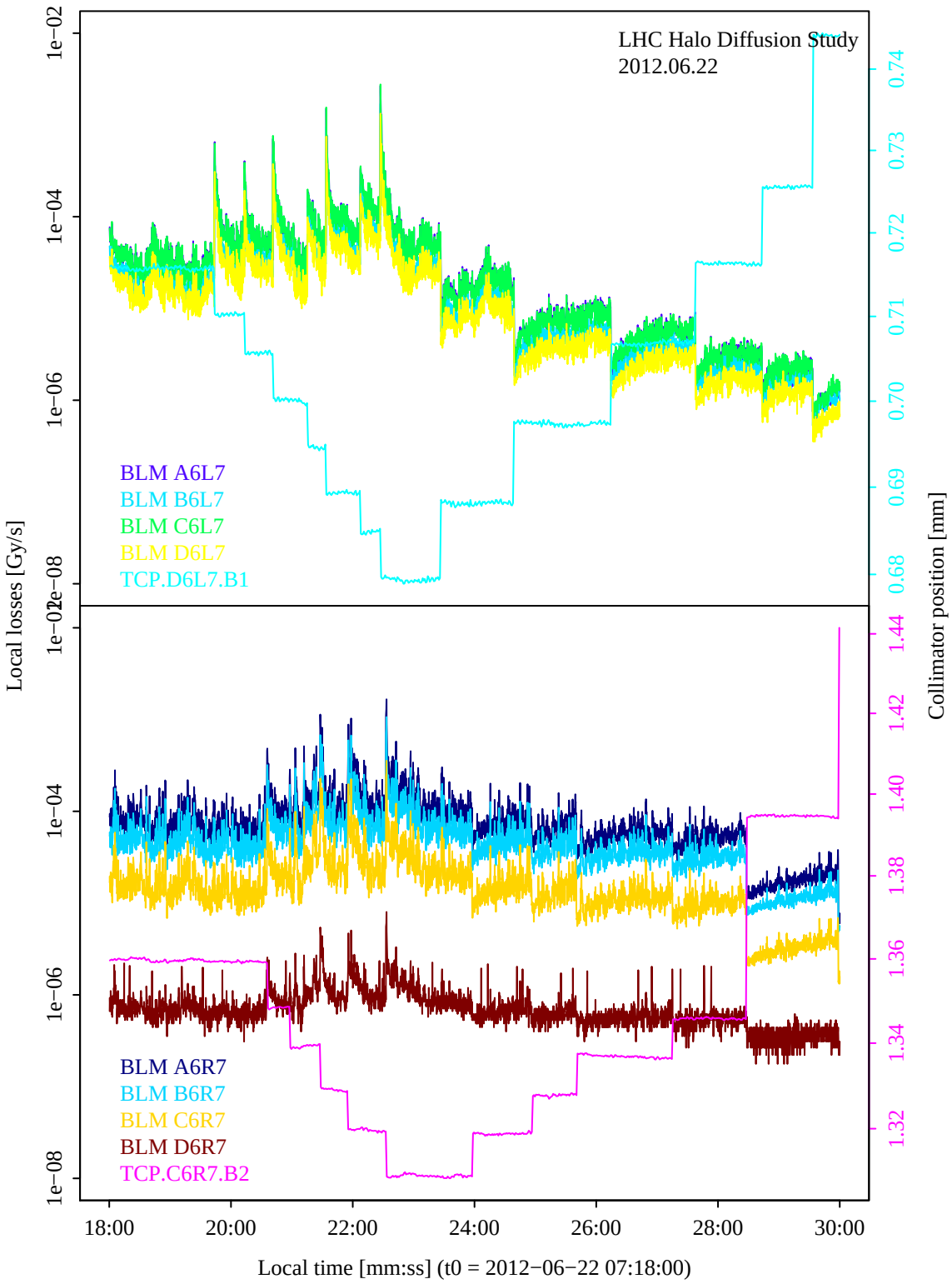


FIG. 7. Comparison of local loss monitors for a portion of the collimator scan.

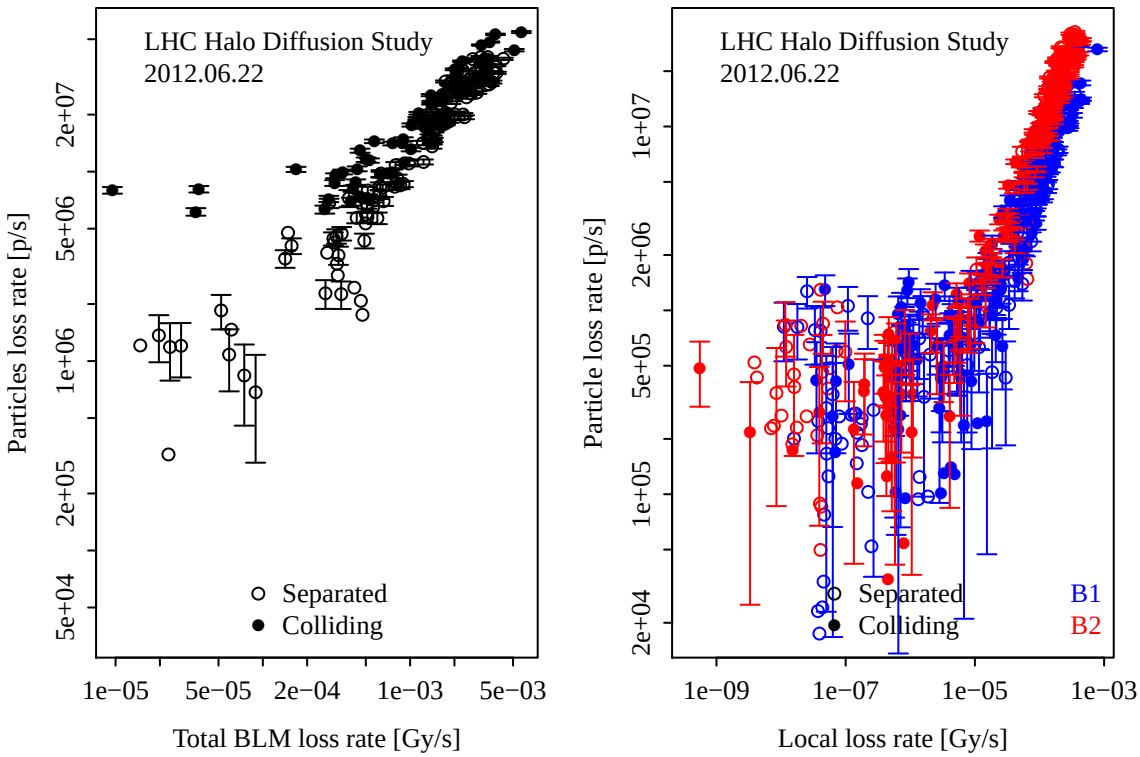


FIG. 8. Particle loss rates vs. BLM loss rates: total (all BLMs, both beams) on the left; divided by beam (only local primary BLMs) on the left.

above) appear to scale linearly with proton losses when the collimator of interest is the leading jaw.

Figures 9 and 10 show the variation of the BLM calibration constant as a function of collimator half gap. During each 20-s interval, the average collimator position is recorded. Measurements are then grouped in 40-μm half-gap intervals. The central value and the spread are shown in the plots.

The calibration factor, or "local loss detection efficiency", varies by more than 2 orders of magnitude. It clearly exposes the threshold when the collimator becomes the leading edge — a half gap of 0.8 mm in the vertical plane and 1.5 mm horizontally. As expected, the efficiency in collisions is lower, when a larger beam fraction is lost at the collider detectors. This is particularly obvious in the horizontal case.

For the purposes of the diffusion analysis, these numbers are an experimental measurement of the k parameter in Eq. 1. In Figures 9 and 10, the measurements of k as a function of collimator position are interpolated with a smooth spline with 4 degrees of freedom. This empirical interpolation is used in Section V to extract the values of the particle fluxes (DA_i, DA_f) and beam distribution slopes (A_i, A_f).

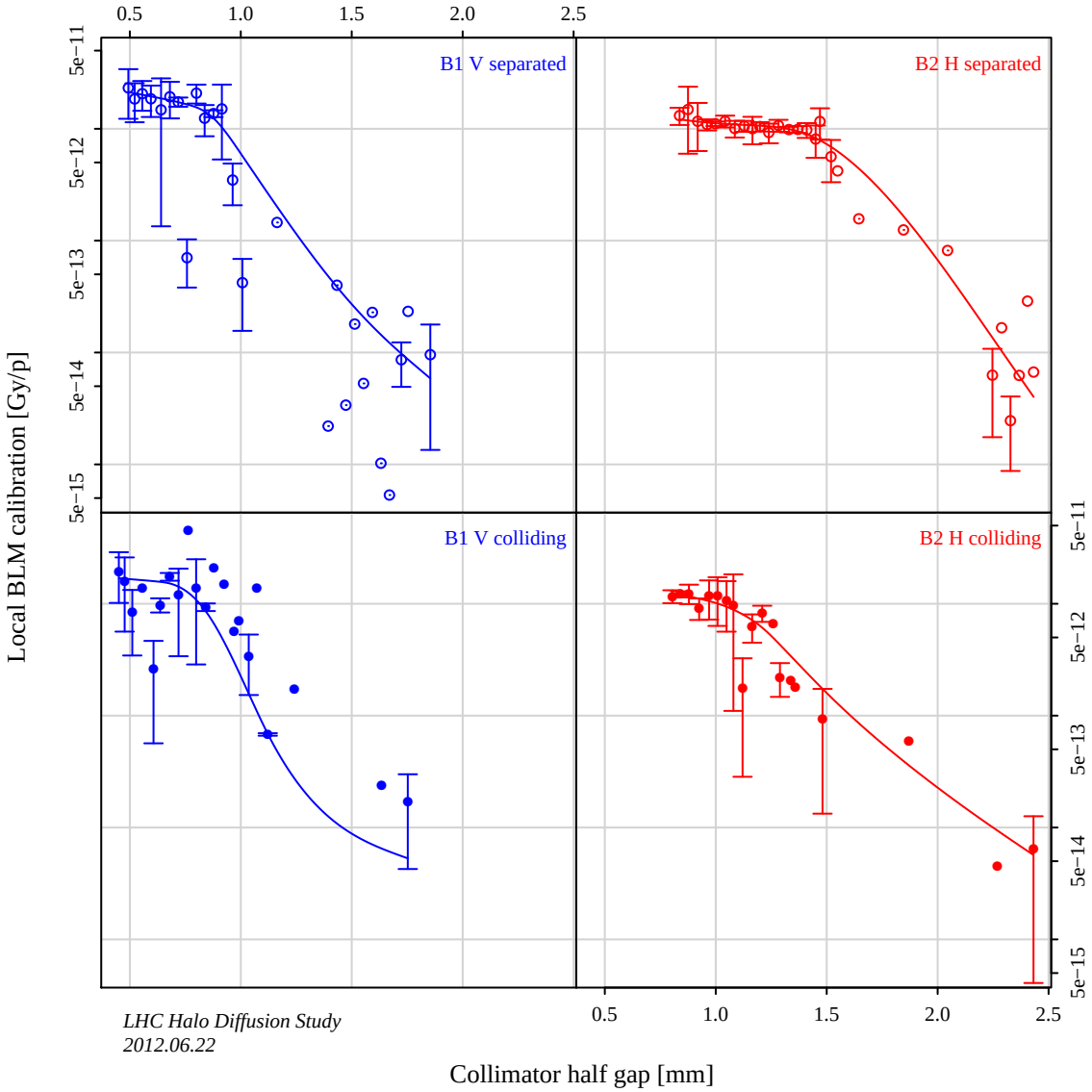


FIG. 9. Local loss monitor calibration vs. collimator position (log scale).

The initial emittances are calculated from the wire-scan and the synchrotron-light data. Both wire scan and sync-light emittance measurements are affected by the scraping of the tails (Figure 11). Emittances before 2012-06-22 06:08:00 (separated beams, before shaving; dashed line in Figure 11) are used to convert collimator gaps into beam sigmas. The first wire-scan measurements are used (first line in Table II). The statistical uncertainty is calculated from the difference between inward and outward fly. The average of the sync-light measurements is used, with a statistical uncertainty deduced from the spread in data points. The final emittance value is obtained by averaging wire scan and synch-light data. Its error is the combination in quadrature of the statistical errors plus a systematic error due to the difference between the two techniques. Results

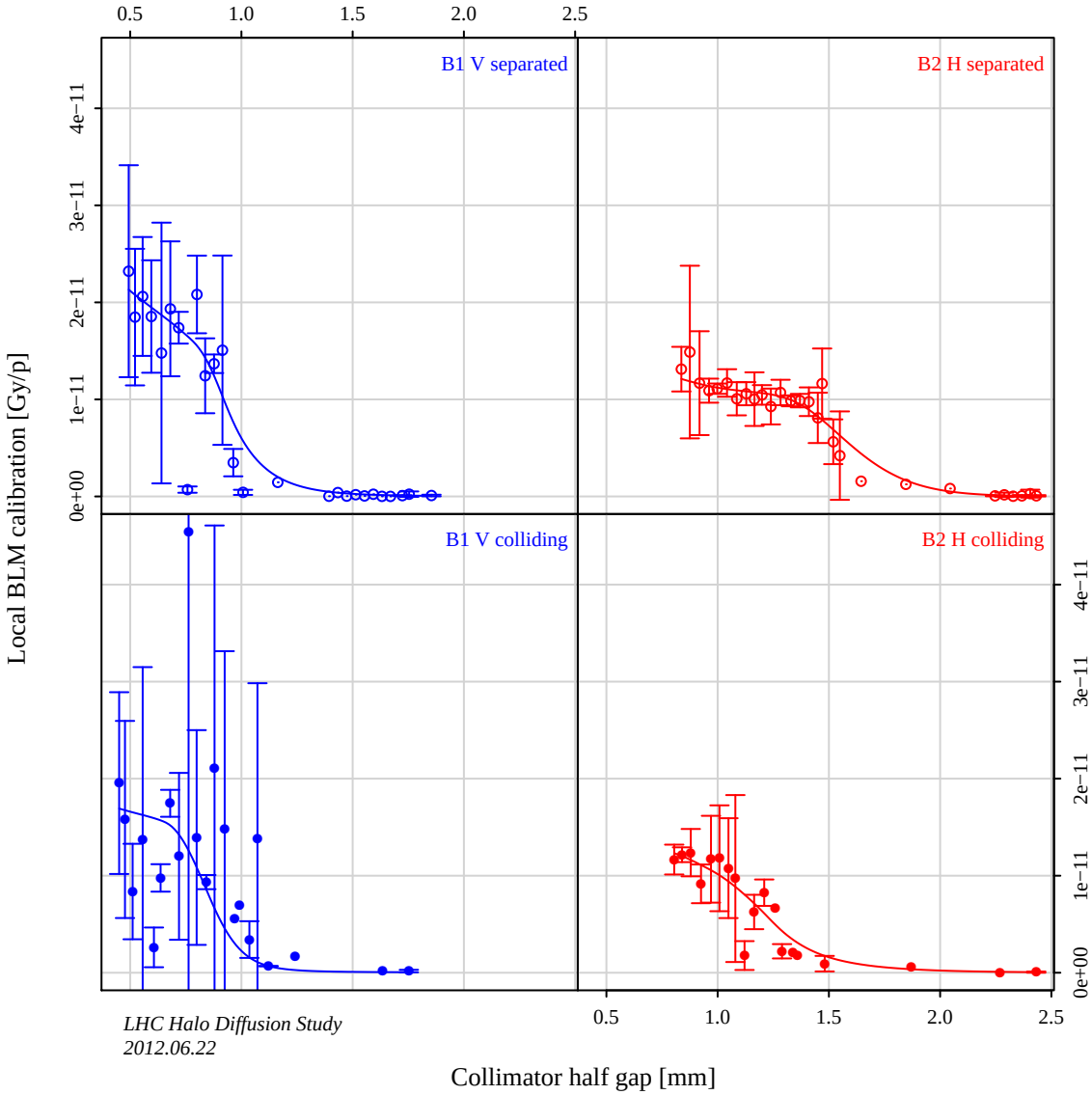


FIG. 10. Local loss monitor calibration vs. collimator position (linear scale).

are reported in Table III. The beam sizes at the collimators calculated from the initial emittances are 0.156 mm (B1 vertical) and 0.270 mm (B2 horizontal).

The emittance growth rates $\gamma \equiv \dot{\varepsilon}/\varepsilon$ are deduced from the slope of the sync-light data before 2012-06-22 06:08:00 (Figure 11, red lines). From them, the expected slope of the diffusion coefficient in the beam core is calculated: $D' \equiv dD/dJ = \dot{\varepsilon}$ (geometrical emittance). These numbers are also reported in Table III and will be compared with the measured diffusion rates.

Collimator steps are defined within the experiment intervals: between 2012-06-22 05:48:00 and 2012-06-22 06:47:00 for separated beams, and between 2012-06-22 06:58:00 and 2012-06-22 07:42:00 for colliding beams. Collimator steps and their properties are listed in Table IV in Appendix B.

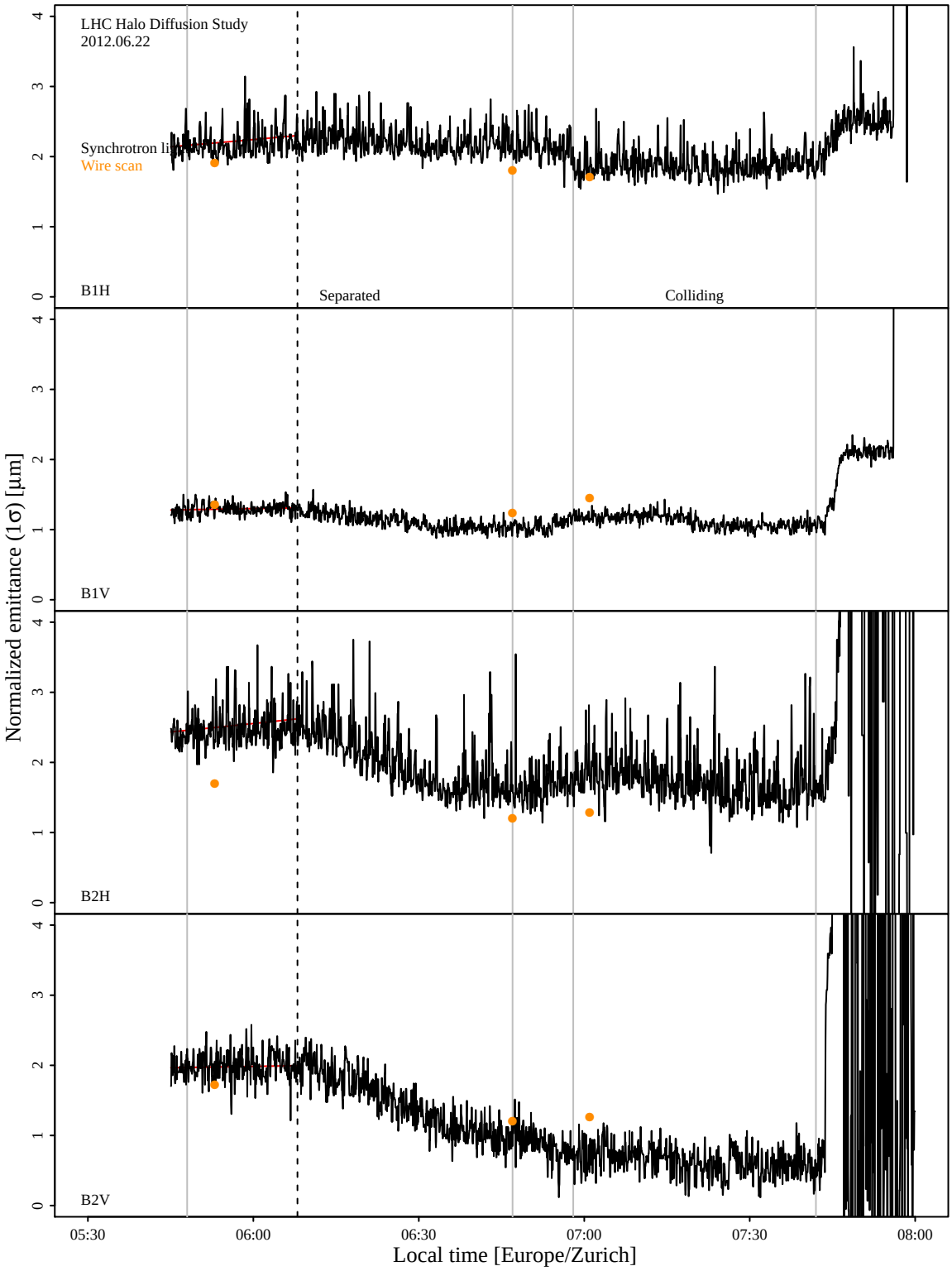


FIG. 11. Synchrotron-light and wire-scan emittances over the course of the study.

	ε_x^{B1} [μm]	ε_y^{B1} [μm]	ε_x^{B2} [μm]	ε_y^{B2} [μm]	Notes
05:53	1.92	1.36	1.69	1.73	Squeezed, separated beams, before scrape
06:47	1.8	1.23	1.2	1.21	Squeezed, separated beams, after scrape
07:01	1.71	1.45	1.28	1.26	Colliding beams, before scrape

TABLE II. Normalized, 1σ wire-scan emittances.

	ε_{ws} [μm]	$\delta\varepsilon_{ws}$ [μm]	ε_{sl} [μm]	$\delta\varepsilon_{sl}$ [μm]	ε [μm]	$\delta\varepsilon$ [μm]	γ [1/s]	$\delta\gamma$ [1/s]	D' [μm/s]	$\delta D'$ [μm/s]
B1.h	1.92	0.013	2.15	0.0096	2.03	0.07	5.23e-05	1.5e-05	2.49e-08	7.4e-09
B1.v	1.36	0.002	1.29	0.0047	1.33	0.02	1.7e-05	1e-05	5.27e-09	3.2e-09
B2.h	1.69	0.0066	2.43	0.013	2.06	0.21	5.37e-05	1.9e-05	2.6e-08	9.6e-09
B2.v	1.73	0.046	1.98	0.012	1.85	0.087	1.1e-05	1.7e-05	4.79e-09	7.2e-09

TABLE III. Initial emittances (wire scan, synchrotron light, average), emittance growth rates γ, and core diffusion slopes D'.

The model described in Section II is used to describe the step data and to extract its main parameters: the diffusion coefficients D and the steady-state loss rates before and after the step, kDA_i and kDA_f, respectively.

Initial values for the parameters are determined as follows. The steady-state rates are evaluated by averaging losses before the step (-0.5 s from the step or earlier) and after (4 s after or later). The starting point for the diffusion coefficient is the value obtained from the decay 3 s after the step (losses $\approx 1/\sqrt{t}$). The time of the step, t_0, is known from the time stamps of the collimator position (1 s resolution) and from the increase in losses (1/12.5 Hz resolution). The duration of the step Δt is estimated from the step size and the nominal jaw speed, 2 mm/s. (This may not be accurate for small steps.) Step time and duration can also be left as free parameters within some reasonable limits.

The distribution of the response variable (loss rate) is not Gaussian, as one can see for instance from Figure 7, because of instrumental effects (such as beam jitter and mechanical vibrations) and because of the Poissonian nature of the shower process. For this reason, fit parameters are determined by minimizing not a χ^2, but the sum of absolute deviations of the model from the data (robust estimation). For the same reson, a first estimate of the statistical errors comes from the Hessian matrix of the χ^2 function, but the final error analysis is done by bootstrapping. For each step, bootstrapping is done by resampling the data points with replacement and repeating the fit

18 times with the same initial parameters. The final results are the median of the fit results and the uncertainties come from their spread.

From the diffusion model fits to the loss rate evolution, one can calculate the particle fluxes

$$\phi = (kDA)/k, \tag{4}$$

where the parameter (kDA) comes from the fit, and the value of k is taken from the loss rate calibration (Figures 9 and 10).

In the diffusion model, the steady-state beam density distribution ($\partial_t f = 0$) for a given collimator position is a function of the particle flux at the collimator and of the diffusion coefficient as a function of amplitude:

$$f(J_a, J) = \phi_a \int_J^{J_a} \frac{1}{D(J')} dJ'. \tag{5}$$

V. RESULTS

A. Diffusion Coefficients

The results of the fits are plotted in Appendix A and listed in Table V of Appendix C. Not all fits converged during this preliminary analysis. Moreover, in some cases, the model cannot explain the first few seconds of losses after the step. This seems to be due to the losses decaying differently from $1/\sqrt{t}$, as shown in the first column of diagnostic plots (Appendix A). This fact could be a combination of instrumental effects (the details of the collimator movement) and pure beam removal/cleaning. But, in general, by looking at all 239 steps, the model seems to reproduce the main features of the loss evolution with time quite well.

The diffusion coefficients as a function of action are shown in Figure 12 for all 4 cases: horizontal and vertical, separated and colliding beams. In the horizontal plane, there seems to be little difference between the separated and colliding cases. In the vertical plane, collisions enhance diffusion by about 2 orders of magnitude. A significant systematic effect are the larger diffusion rates obtained from the outward steps. They are not understood, but are probably due to the fact that, after scraping, a different, quickly-diffusing beam population is being sampled.

The grey curves in Figure 12 are not an interpolation of the data: they represent the dependence $D(J)$ that one would obtain from the core emittance growth rates (Table III). This comparison allows one to draw several conclusions: (1) the diffusion rates measured with the collimator scan technique have reasonable values; (2) without collisions, the beam halo in the LHC diffuses almost

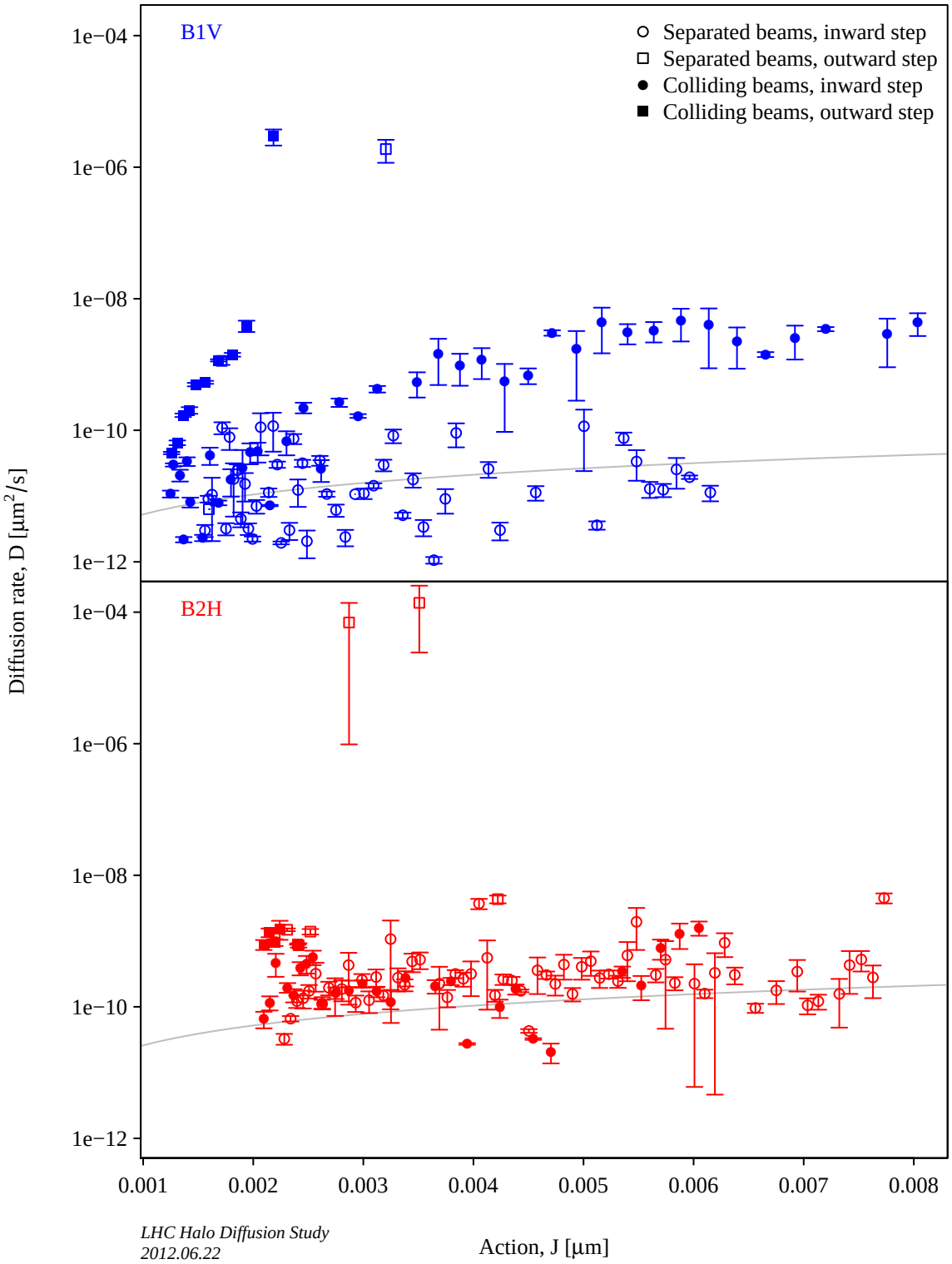


FIG. 12. Diffusion coefficient vs. action from the collimator scan (points) compared with the expectation from core emittance growth rates (lines).

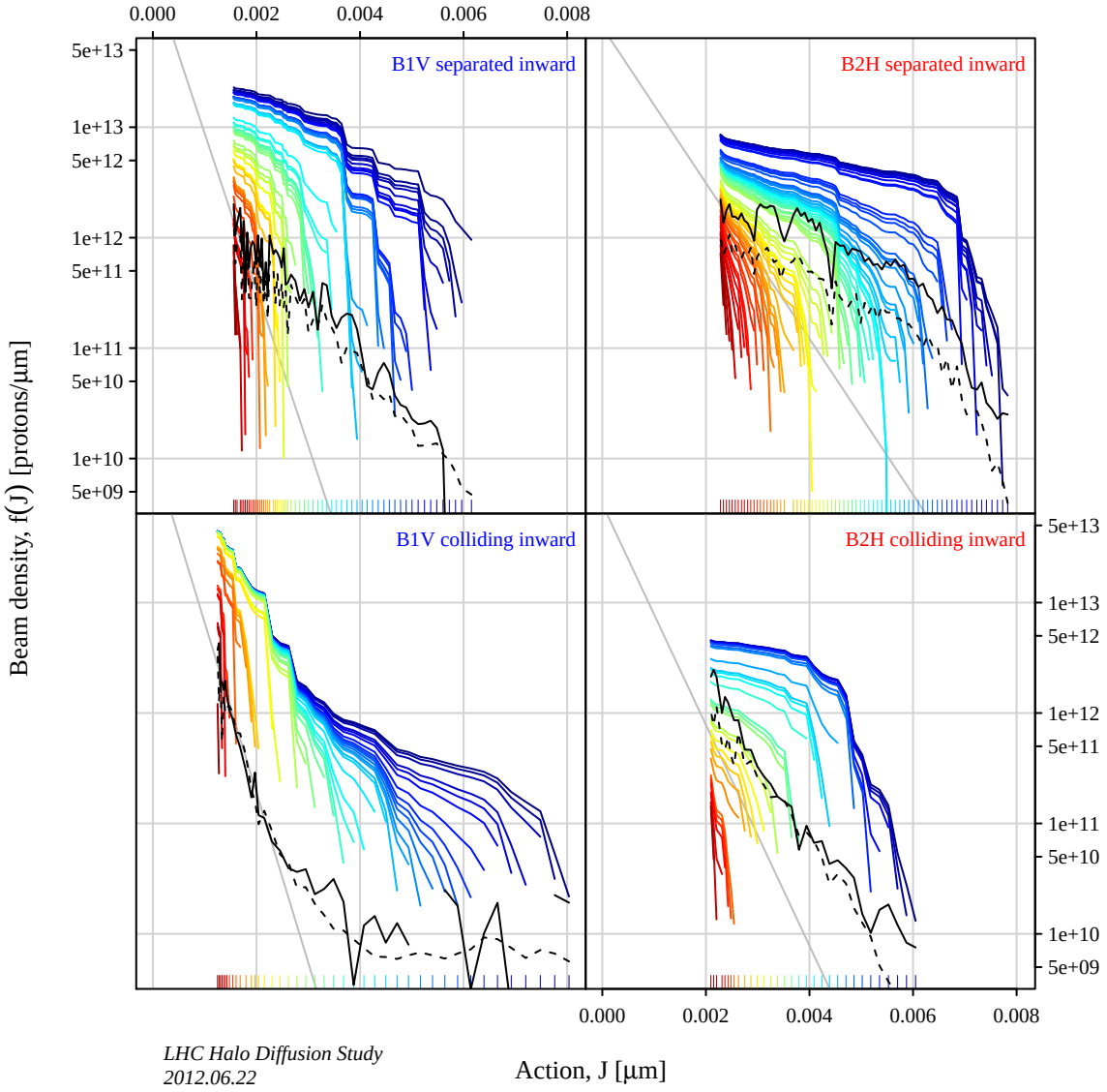


FIG. 13. Calculated beam distributions as a function of collimator position during the inward scans: from intensity loss (solid black); from integrated losses (dashed black); from the diffusion coefficient, unnormalized (from blue to red). The gray line denotes a Gaussian core with the measured beam emittance. The colored vertical tick marks indicate the corresponding collimator positions.

like the core up to large amplitudes; (3) no significant dynamic aperture effects (sharp rises in diffusion rate) are observed in the amplitude range explored in these experiments.

B. Halo Population

In the approximation of static beam distributions, the beam densities can be calculated from the measured intensity loss during a short interval (4 s in this analysis) centered around the collimator

movement. The results are shown in Figure 13 (solid black lines). Similar results are obtained by integrating the calibrated losses over the same short interval (Figure 13, dashed black lines). For comparison, a Gaussian core with the measured beam emittance is also shown in Figure 13 (gray line). It is interesting to note the depletion in the case of collisions compared with separated beams.

In the diffusion model, beam distributions evolve with the collimator position. The collimator determines the boudary condition for the density ($f(J_c) = 0$). Densities as a function of collimator position are calculated according to Eq. 5. Figure 13 shows the results (not normalized, for the moment). One can see how the distribution is cut as the collimators move in. Apart from the arbitrary normalization factor, the initial distributions (in blue) exibit trends similar to the ones deduced from intensities and losses.

C. Impact parameter distributions

To do: impact parameter distribution from diffusion coefficients, following Ref. [12].

D. Beam Jitter

The time structure of losses with tight collimator settings is a sensitive probe of beam vibrations. As an example, Figure 14 shows the fast Fourier transform spectrum of losses when the collimator positions were closer than 3.75 σ. In the horizontal plane, a sharp line around 5.8 Hz is visible. The vertical plane shows some activity around 2 Hz and a prominent peak at 4.6 Hz. In the Tevatron, the latter peak was also present. It was attributed to mechanical vibrations of the compressors in the Central Helium Liquefier. Does the LHC cryogenic system operate at similar frequencies?

By comparing the amplitude of the peaks with the amplitude of the DC component and of the loss spikes for a known collimator step, one may estimate the amplitude of these vibrations. A rough calibration indicates amplitudes of at least a few nanometers at the collimator jaw.

VI. CONCLUSIONS

For the first time in the LHC, transverse beam halo diffusion rates were measured. Separated beam exibited a slow halo diffusion, comparable with the emittance growth from the core. No dynamic aperture effects were observed up to about 7 σ. This can be interpreted as a confirmation

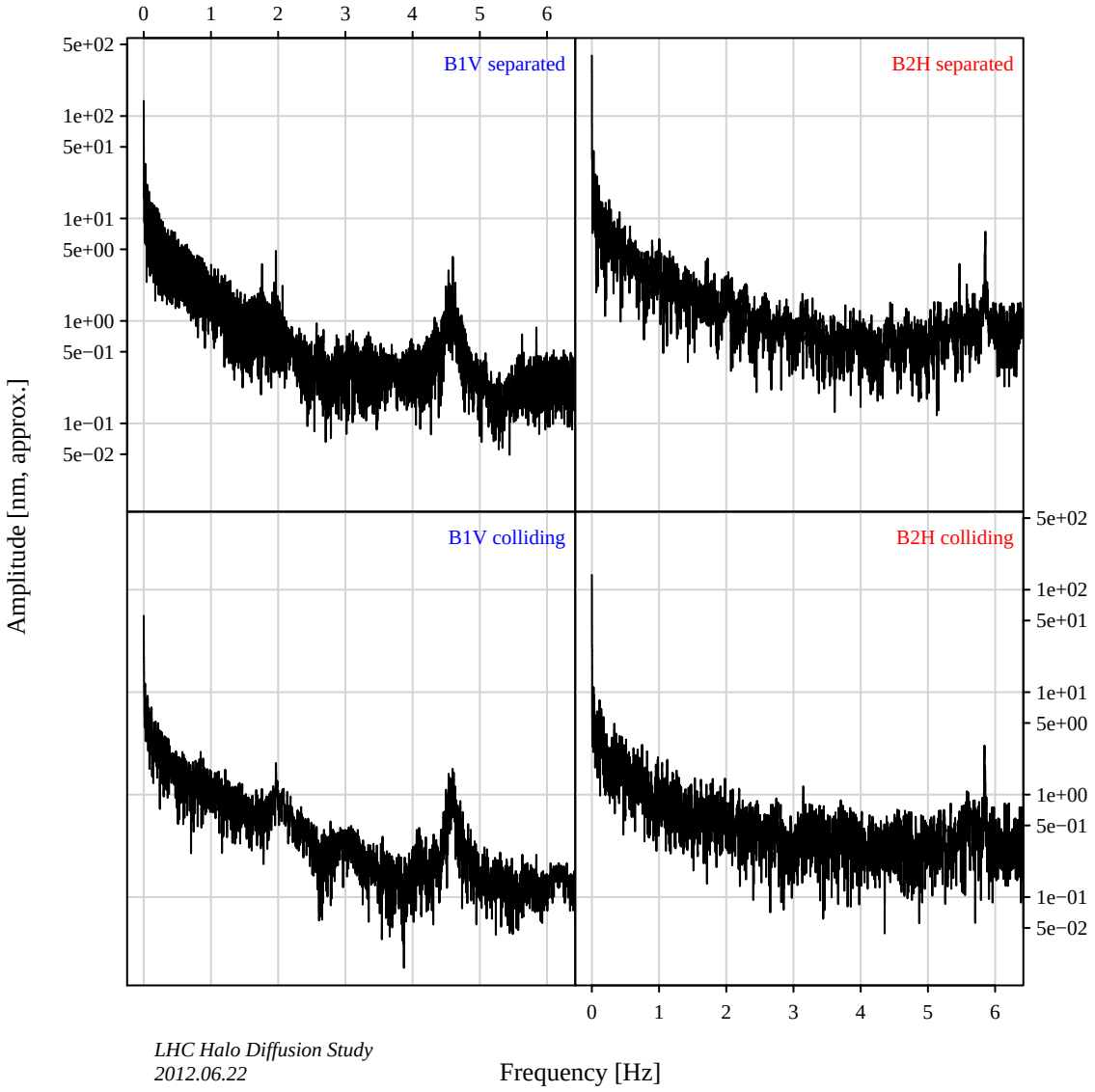


FIG. 14. Beam jitter FFTs.

of the extremely good quality of the magnetic fields in the machine. Collisions enhanced halo diffusion in the vertical plane by about 2 orders of magnitude.

From the measured diffusion coefficents, estimates of beam impact parameters on the primary collimator jaws are possible (work in progress).

Beam population measurements exposed the presence and magnitude of non-Gaussian tails.

The frequency spectrum of losses with tight collimator settings was used as a very sensitive probe of low-frequency beam vibrations, showing clear indications of small beam jitter of a few nanometers in amplitude.

These measurements shed light on the relationship between halo population and dynamics,

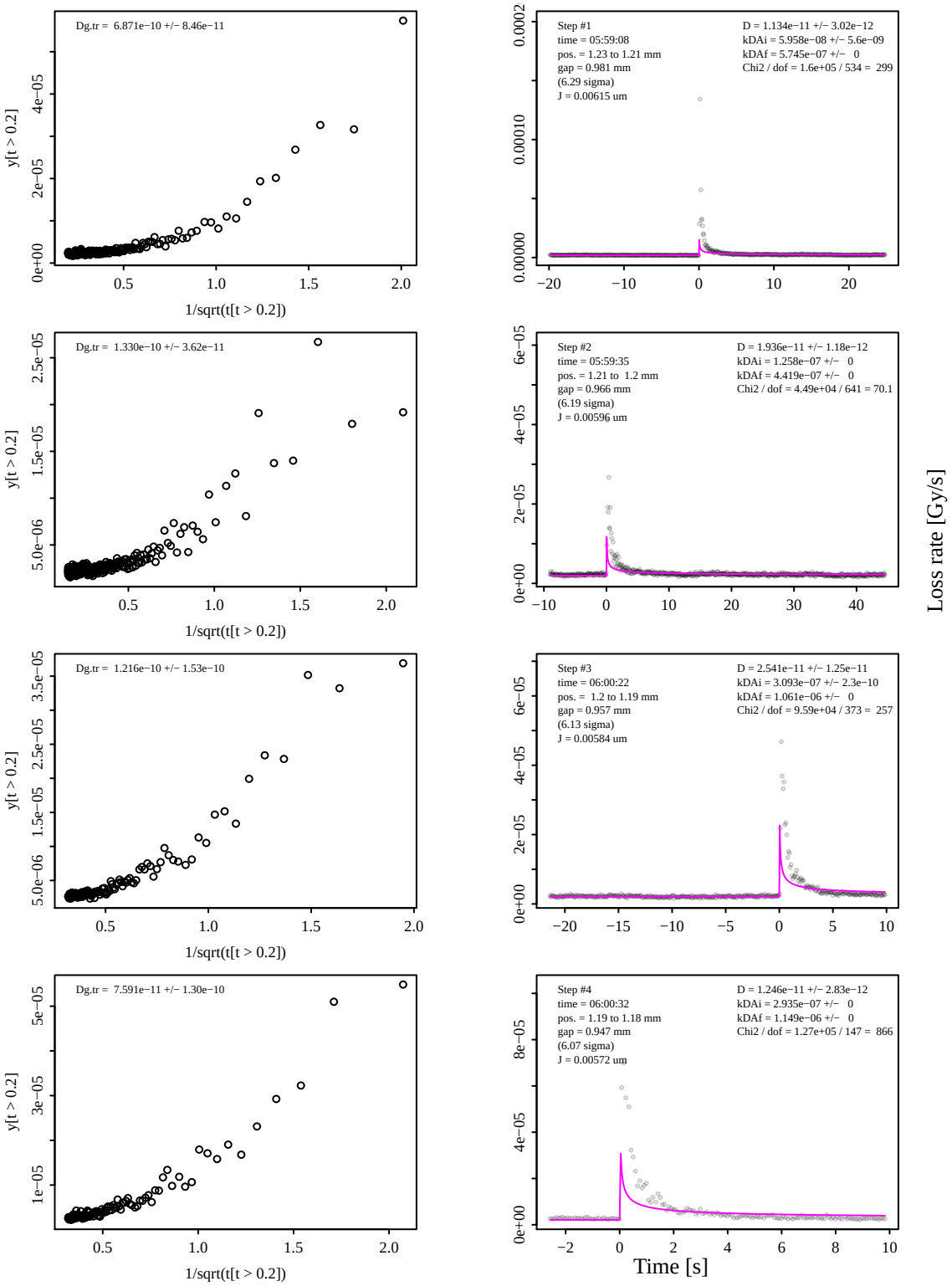
emittance growth, beam lifetime, and collimation efficiency. They are also important inputs for collimator system design and upgrades, including new methods such as crystal channeling or hollow electron lenses.

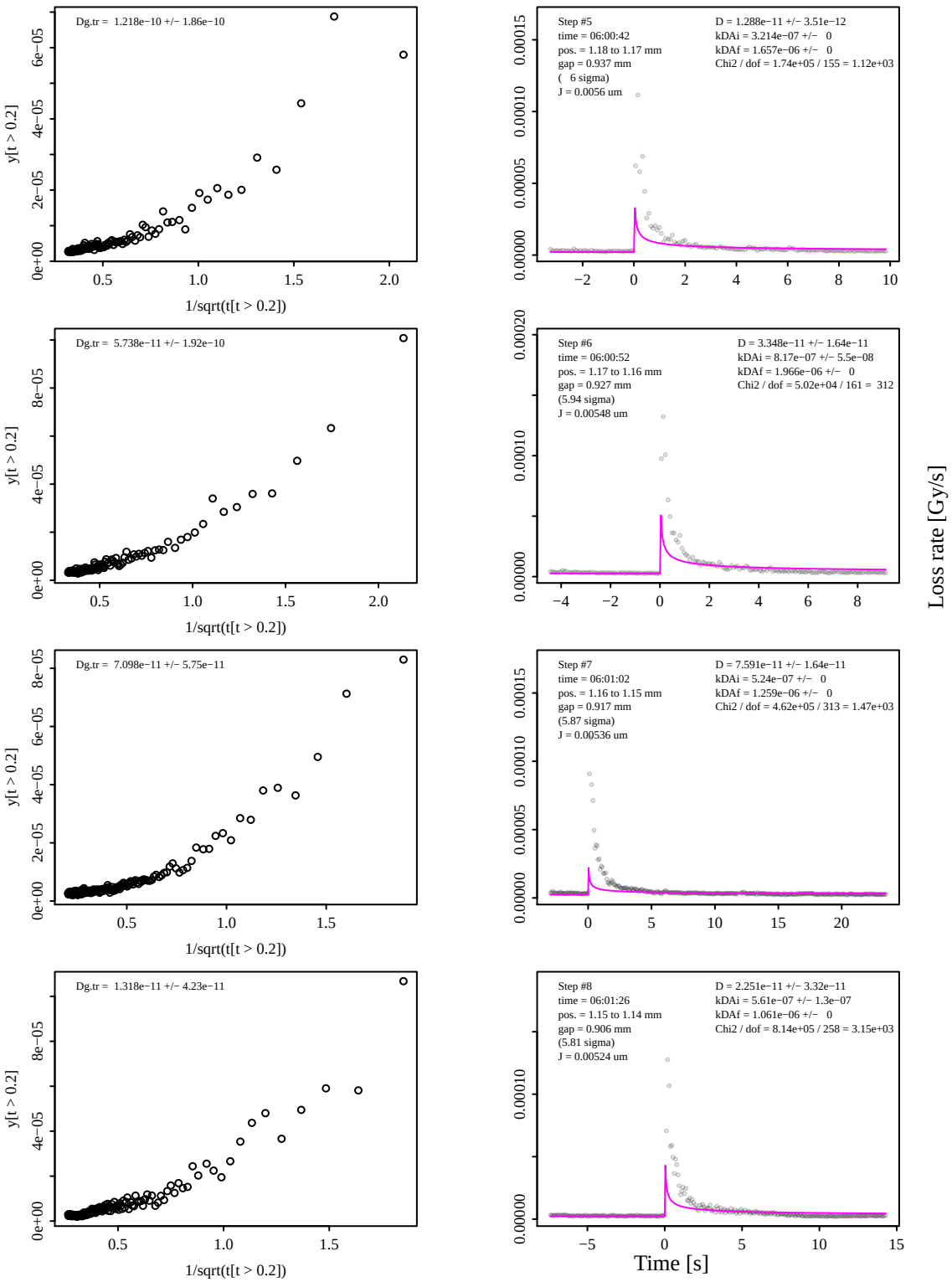
VII. ACKNOWLEDGMENTS

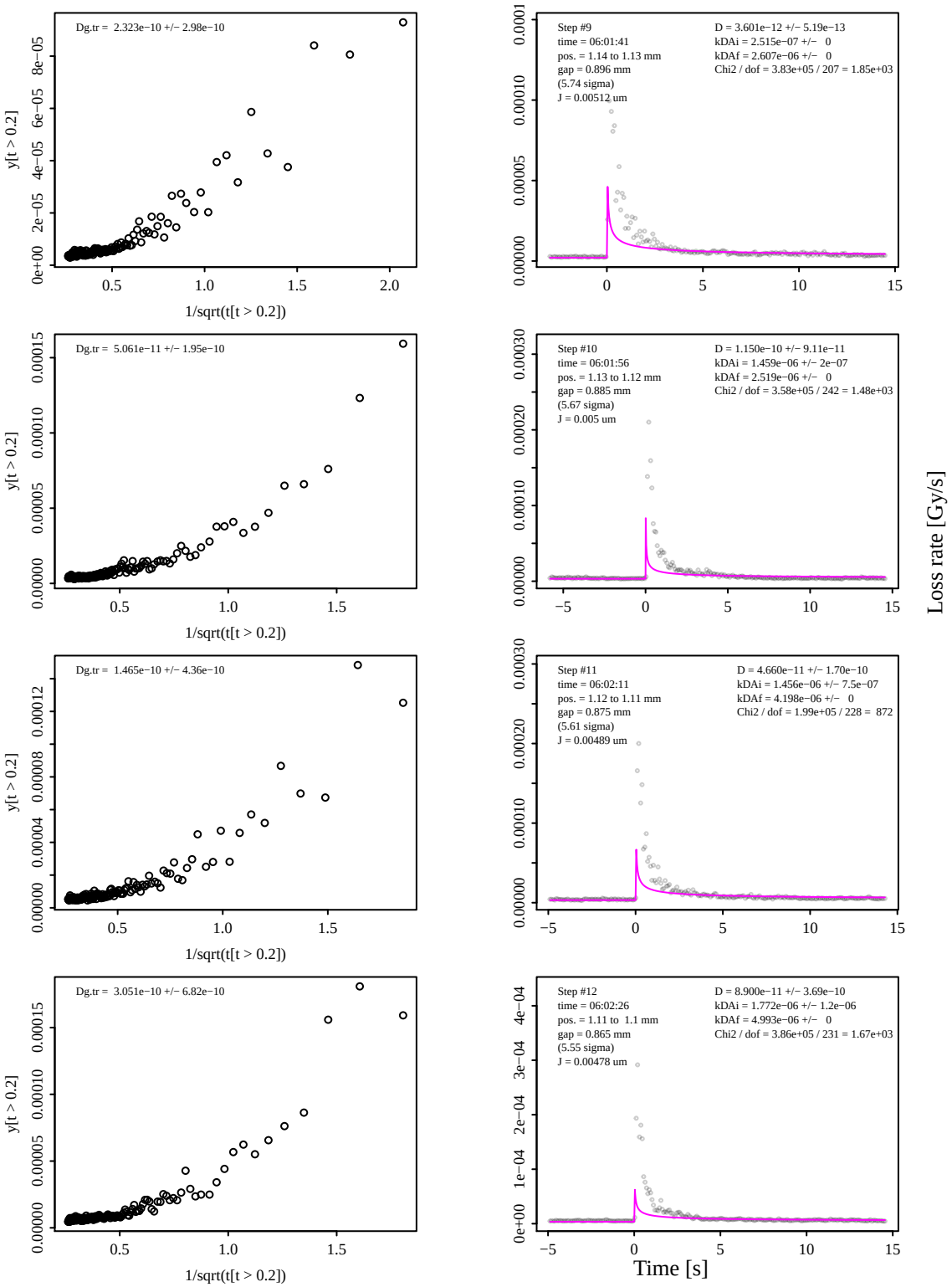
These measurements would not have been possible without the help of the LHC operators on shift during the experiments: S. Cettour Cave, A. MacPherson, D. Jacquet, and M. Solfaroli. R. De Maria provided the software infrastructure to acquire beam current data at high frequency.

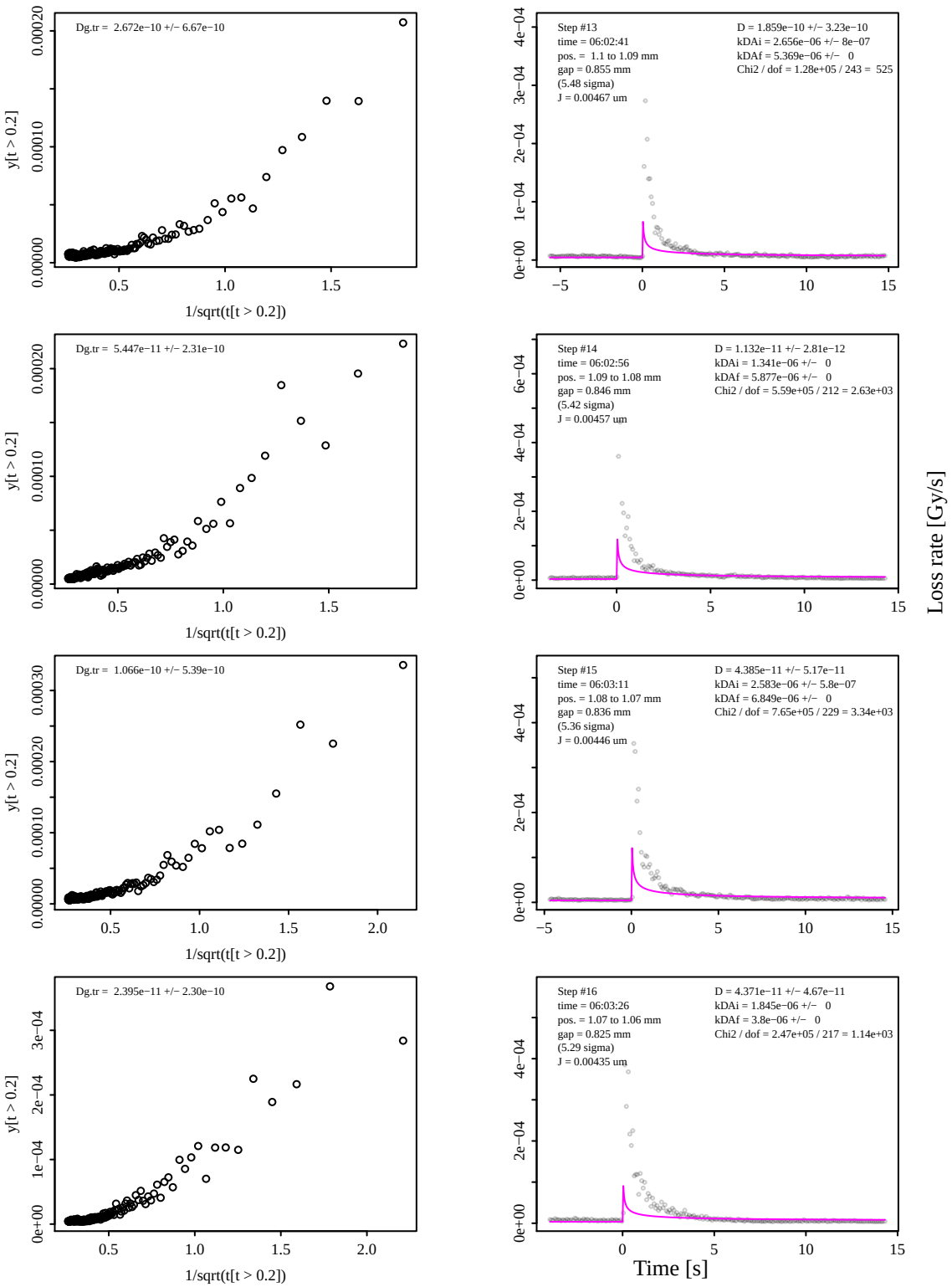
Fermi Research Alliance, LLC operates Fermilab under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.

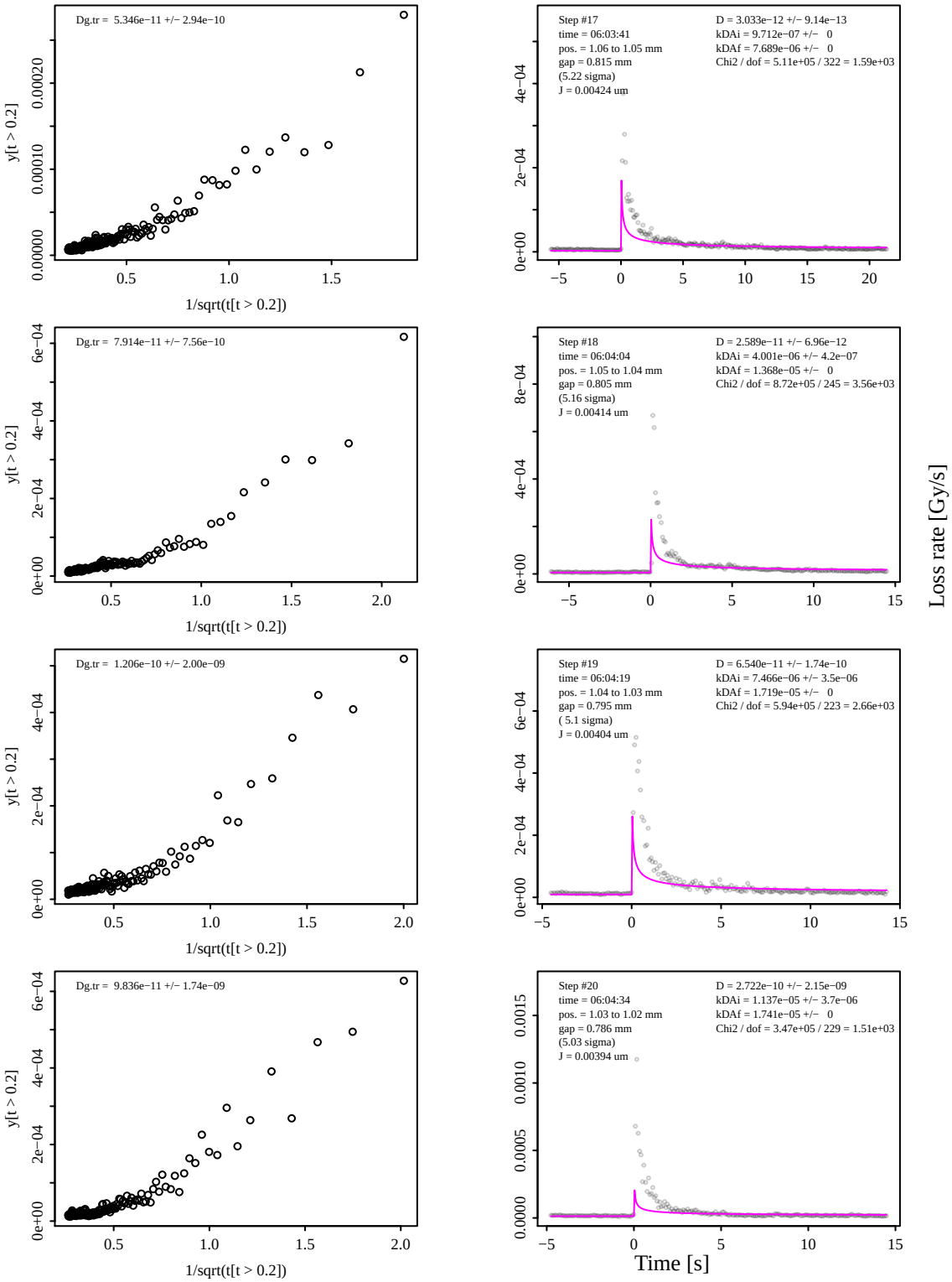
Appendix A: Diagnostic plots

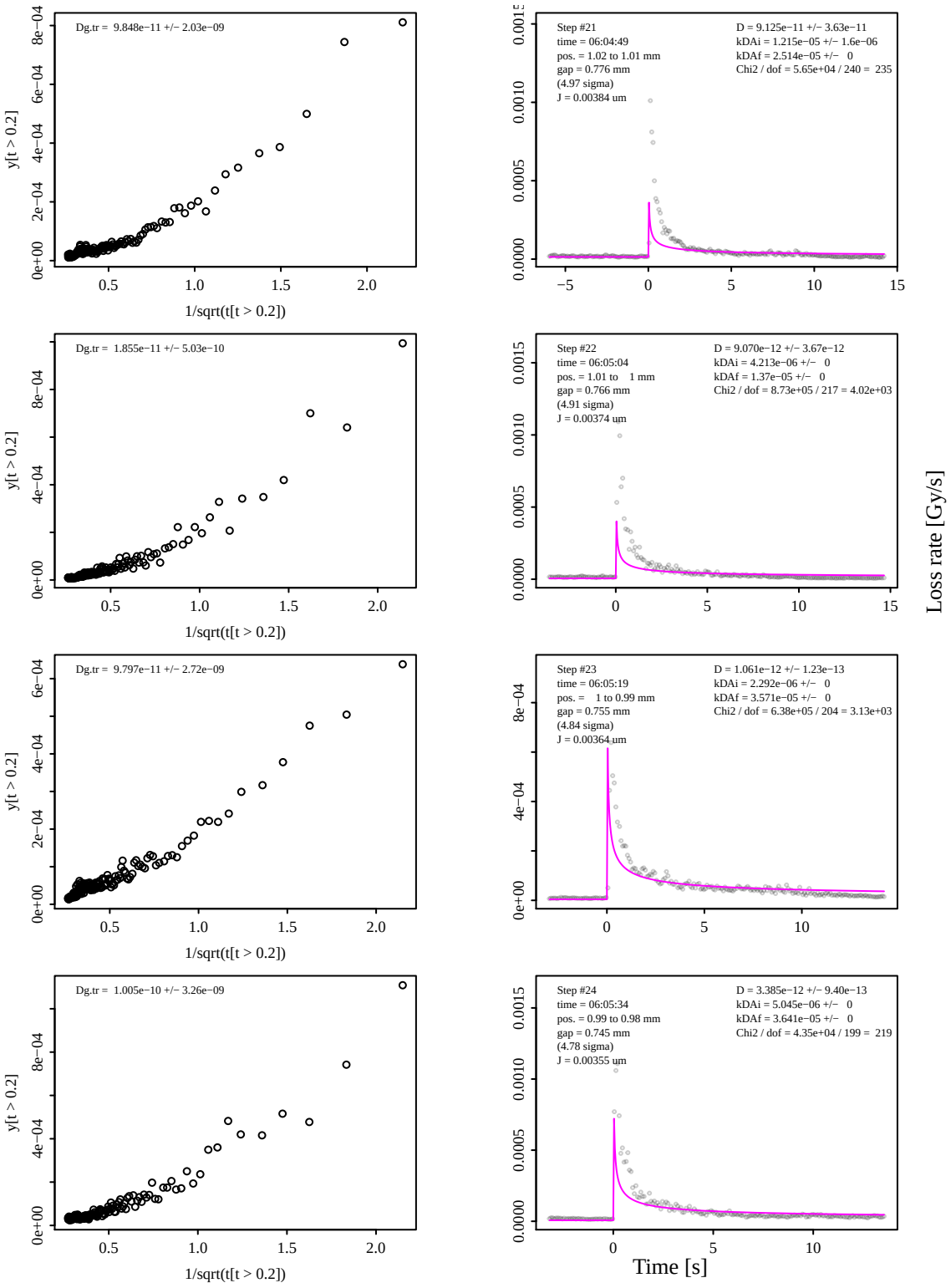


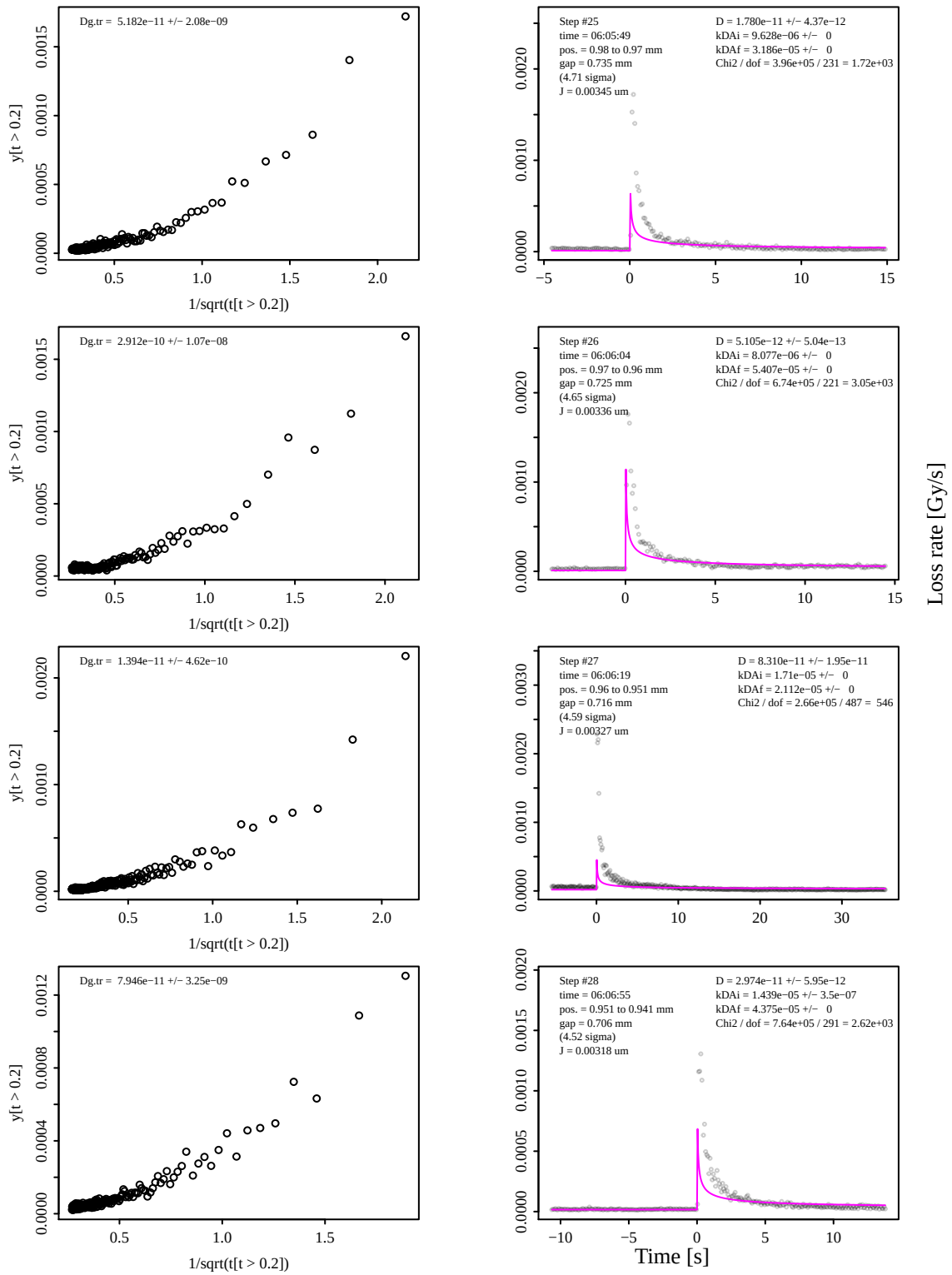

Dg.tr = 1.218e−10 +/− 1.86e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #5
time = 06:00:42
pos. = 1.18 to 1.17 mm
gap = 0.937 mm
(6 sigma)
J = 0.0056 um
D = 1.288e−11 +/− 3.51e−12
kDAi = 3.214e−07 +/− 0
kDAf = 1.657e−06 +/− 0
Chi2 / dof = 1.74e+05 / 155 = 1.12e+03
Dg.tr = 5.738e−11 +/− 1.92e−10
Step #6
time = 06:00:52
pos. = 1.17 to 1.16 mm
gap = 0.927 mm
(5.94 sigma)
J = 0.00548 um
D = 3.348e−11 +/− 1.64e−11
kDAi = 8.17e−07 +/− 5.5e−08
kDAf = 1.966e−06 +/− 0
Chi2 / dof = 5.02e+04 / 161 = 312
Dg.tr = 7.098e−11 +/− 5.75e−11
Step #7
time = 06:01:02
pos. = 1.16 to 1.15 mm
gap = 0.917 mm
(5.87 sigma)
J = 0.00536 um
D = 7.591e−11 +/− 1.64e−11
kDAi = 5.24e−07 +/− 0
kDAf = 1.259e−06 +/− 0
Chi2 / dof = 4.62e+05 / 313 = 1.47e+03
Dg.tr = 1.318e−11 +/− 4.23e−11
Step #8
time = 06:01:26
pos. = 1.15 to 1.14 mm
gap = 0.906 mm
(5.81 sigma)
J = 0.00524 um
D = 2.251e−11 +/− 3.32e−11
kDAi = 5.61e−07 +/− 1.3e−07
kDAf = 1.061e−06 +/− 0
Chi2 / dof = 8.14e+05 / 258 = 3.15e+03
Loss rate [Gy/s]
Time [s]

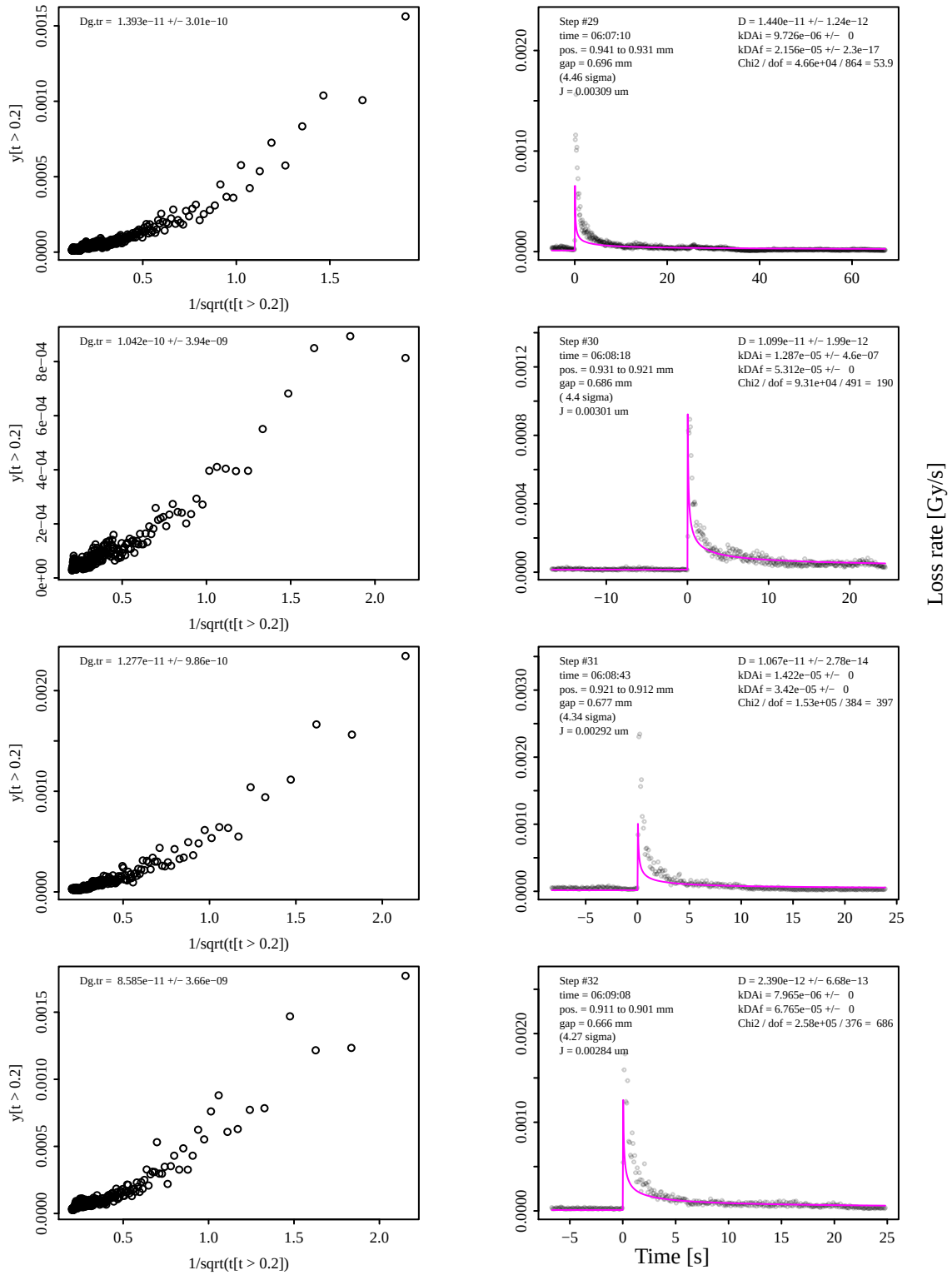

Dg.tr = 2.323e−10 +/− 2.98e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #9
time = 06:01:41
pos. = 1.14 to 1.13 mm
gap = 0.896 mm
(5.74 sigma)
J = 0.00512 um
D = 3.601e−12 +/− 5.19e−13
kDAi = 2.515e−07 +/− 0
kDAf = 2.607e−06 +/− 0
Chi2 / dof = 3.83e+05 / 207 = 1.85e+03
Dg.tr = 5.061e−11 +/− 1.95e−10
Step #10
time = 06:01:56
pos. = 1.13 to 1.12 mm
gap = 0.885 mm
(5.67 sigma)
J = 0.005 um
D = 1.150e−10 +/− 9.11e−11
kDAi = 1.459e−06 +/− 2e−07
kDAf = 2.519e−06 +/− 0
Chi2 / dof = 3.58e+05 / 242 = 1.48e+03
Dg.tr = 1.465e−10 +/− 4.36e−10
Step #11
time = 06:02:11
pos. = 1.12 to 1.11 mm
gap = 0.875 mm
(5.61 sigma)
J = 0.00489 um
D = 4.660e−11 +/− 1.70e−10
kDAi = 1.456e−06 +/− 7.5e−07
kDAf = 4.198e−06 +/− 0
Chi2 / dof = 1.99e+05 / 228 = 872
Dg.tr = 3.051e−10 +/− 6.82e−10
Step #12
time = 06:02:26
pos. = 1.11 to 1.1 mm
gap = 0.865 mm
(5.55 sigma)
J = 0.00478 um
D = 8.900e−11 +/− 3.69e−10
kDAi = 1.772e−06 +/− 1.2e−06
kDAf = 4.993e−06 +/− 0
Chi2 / dof = 3.86e+05 / 231 = 1.67e+03
Loss rate [Gy/s]
Time [s]

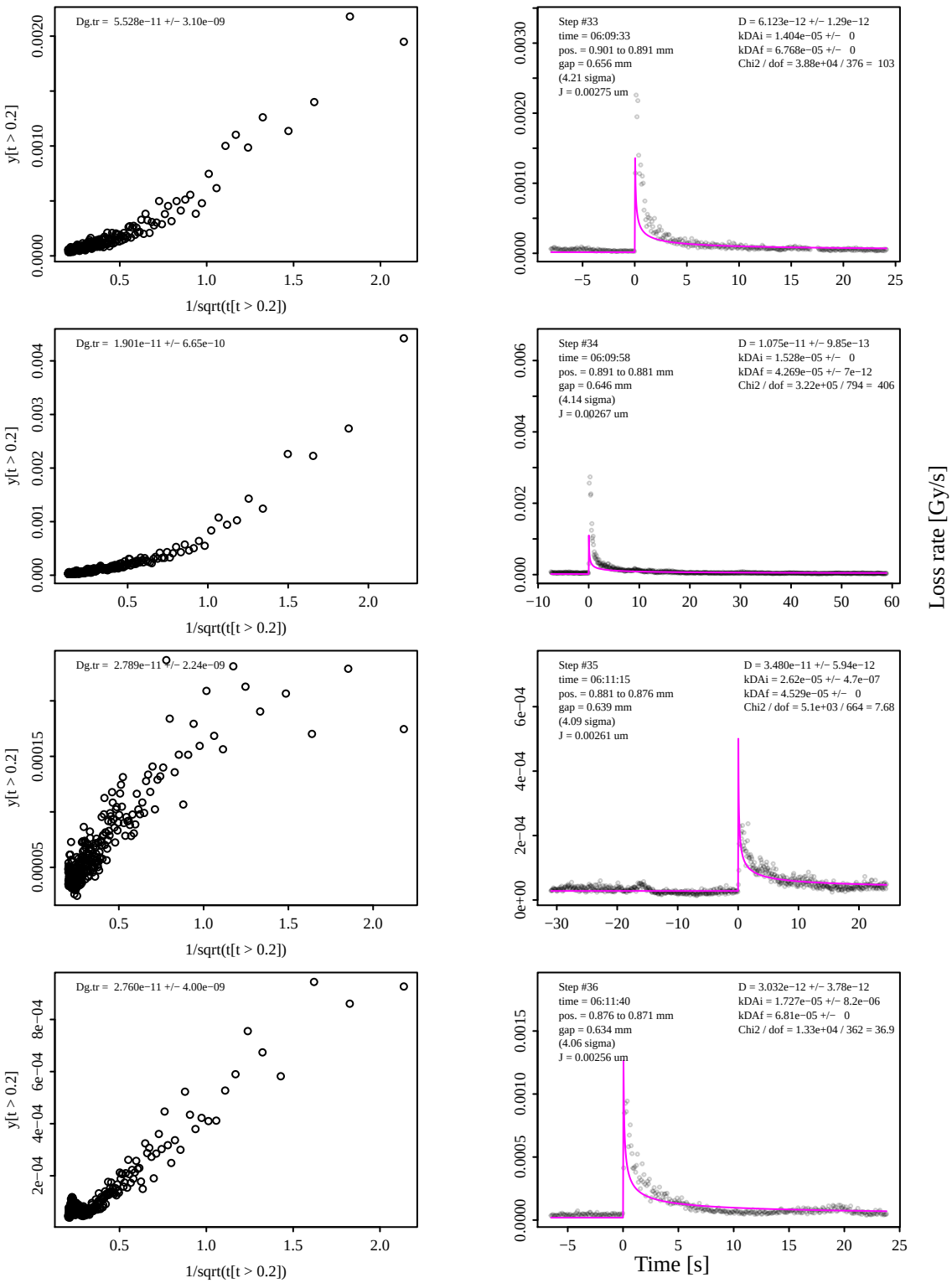

Dg.tr = 2.672e−10 +/− 6.67e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #13
time = 06:02:41
pos. = 1.1 to 1.09 mm
gap = 0.855 mm
(5.48 sigma)
J = 0.00467 um
D = 1.859e−10 +/− 3.23e−10
kDAi = 2.656e−06 +/− 8e−07
kDAf = 5.369e−06 +/− 0
Chi2 / dof = 1.28e+05 / 243 = 525
Dg.tr = 5.447e−11 +/− 2.31e−10
Step #14
time = 06:02:56
pos. = 1.09 to 1.08 mm
gap = 0.846 mm
(5.42 sigma)
J = 0.00457 um
D = 1.132e−11 +/− 2.81e−12
kDAi = 1.341e−06 +/− 0
kDAf = 5.877e−06 +/− 0
Chi2 / dof = 5.59e+05 / 212 = 2.63e+03
Dg.tr = 1.066e−10 +/− 5.39e−10
Step #15
time = 06:03:11
pos. = 1.08 to 1.07 mm
gap = 0.836 mm
(5.36 sigma)
J = 0.00446 um
D = 4.385e−11 +/− 5.17e−11
kDAi = 2.583e−06 +/− 5.8e−07
kDAf = 6.849e−06 +/− 0
Chi2 / dof = 7.65e+05 / 229 = 3.34e+03
Dg.tr = 2.395e−11 +/− 2.30e−10
Step #16
time = 06:03:26
pos. = 1.07 to 1.06 mm
gap = 0.825 mm
(5.29 sigma)
J = 0.00435 um
D = 4.371e−11 +/− 4.67e−11
kDAi = 1.845e−06 +/− 0
kDAf = 3.8e−06 +/− 0
Chi2 / dof = 2.47e+05 / 217 = 1.14e+03
Loss rate [Gy/s]
Time [s]

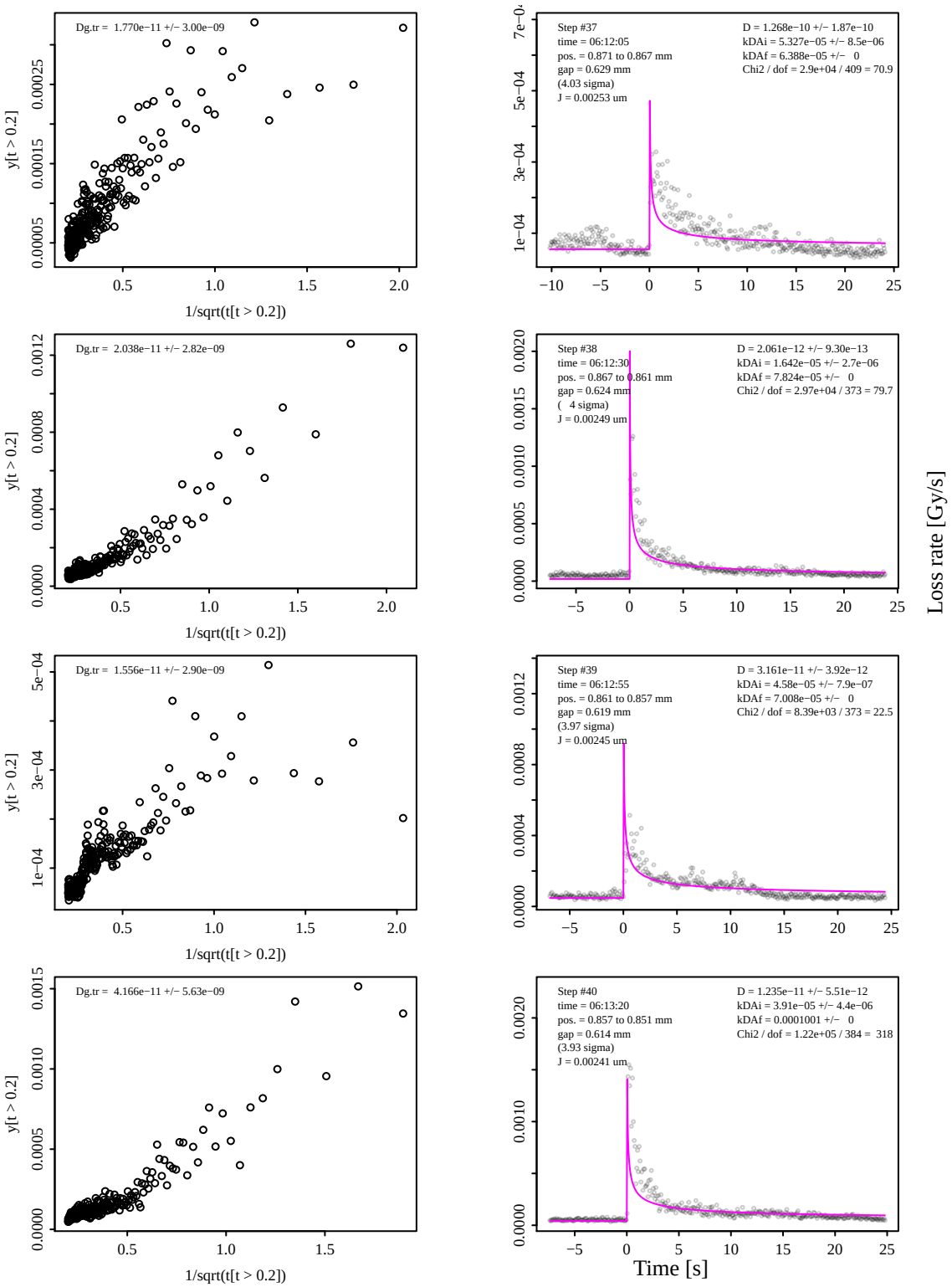

Dg.tr = 5.346e−11 +/− 2.94e−10
Step #17
time = 06:03:41
pos. = 1.06 to 1.05 mm
gap = 0.815 mm
(5.22 sigma)
J = 0.00424 um
D = 3.033e−12 +/− 9.14e−13
kDAi = 9.712e−07 +/− 0
kDAf = 7.689e−06 +/− 0
Chi2 / dof = 5.11e+05 / 322 = 1.59e+03
Dg.tr = 7.914e−11 +/− 7.56e−10
Step #18
time = 06:04:04
pos. = 1.05 to 1.04 mm
gap = 0.805 mm
(5.16 sigma)
J = 0.00414 um
D = 2.589e−11 +/− 6.96e−12
kDAi = 4.001e−06 +/− 4.2e−07
kDAf = 1.368e−05 +/− 0
Chi2 / dof = 8.72e+05 / 245 = 3.56e+03
Dg.tr = 1.206e−10 +/− 2.00e−09
Step #19
time = 06:04:19
pos. = 1.04 to 1.03 mm
gap = 0.795 mm
(5.1 sigma)
J = 0.00404 um
D = 6.540e−11 +/− 1.74e−10
kDAi = 7.466e−06 +/− 3.5e−06
kDAf = 1.719e−05 +/− 0
Chi2 / dof = 5.94e+05 / 223 = 2.66e+03
Dg.tr = 9.836e−11 +/− 1.74e−09
Step #20
time = 06:04:34
pos. = 1.03 to 1.02 mm
gap = 0.786 mm
(5.03 sigma)
J = 0.00394 um
D = 2.722e−10 +/− 2.15e−09
kDAi = 1.137e−05 +/− 3.7e−06
kDAf = 1.741e−05 +/− 0
Chi2 / dof = 3.47e+05 / 229 = 1.51e+03
y[t > 0.2]
1/sqrt(t[t > 0.2])
Loss rate [Gy/s]
Time [s]

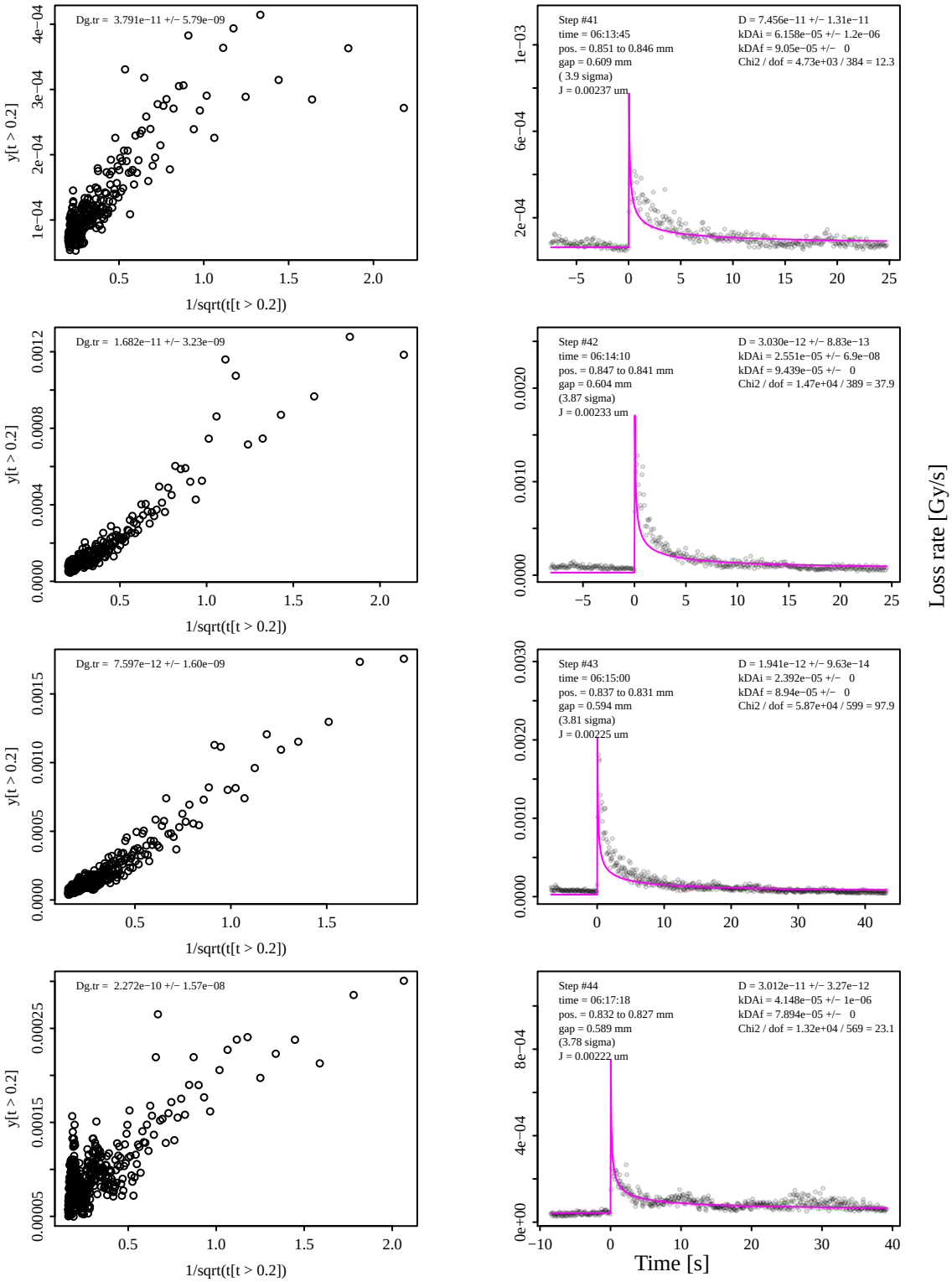

Dg.tr = 9.848e−11 +/− 2.03e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #21
time = 06:04:49
pos. = 1.02 to 1.01 mm
gap = 0.776 mm
(4.97 sigma)
J = 0.00384 um
D = 9.125e−11 +/− 3.63e−11
kDAi = 1.215e−05 +/− 1.6e−06
kDAf = 2.514e−05 +/− 0
Chi2 / dof = 5.65e+04 / 240 = 235
Dg.tr = 1.855e−11 +/− 5.03e−10
Step #22
time = 06:05:04
pos. = 1.01 to 1 mm
gap = 0.766 mm
(4.91 sigma)
J = 0.00374 um
D = 9.070e−12 +/− 3.67e−12
kDAi = 4.213e−06 +/− 0
kDAf = 1.37e−05 +/− 0
Chi2 / dof = 8.73e+05 / 217 = 4.02e+03
Loss rate [Gy/s]
Dg.tr = 9.797e−11 +/− 2.72e−09
Step #23
time = 06:05:19
pos. = 1 to 0.99 mm
gap = 0.755 mm
(4.84 sigma)
J = 0.00364 um
D = 1.061e−12 +/− 1.23e−13
kDAi = 2.292e−06 +/− 0
kDAf = 3.571e−05 +/− 0
Chi2 / dof = 6.38e+05 / 204 = 3.13e+03
Dg.tr = 1.005e−10 +/− 3.26e−09
Step #24
time = 06:05:34
pos. = 0.99 to 0.98 mm
gap = 0.745 mm
(4.78 sigma)
J = 0.00355 um
D = 3.385e−12 +/− 9.40e−13
kDAi = 5.045e−06 +/− 0
kDAf = 3.641e−05 +/− 0
Chi2 / dof = 4.35e+04 / 199 = 219
Time [s]

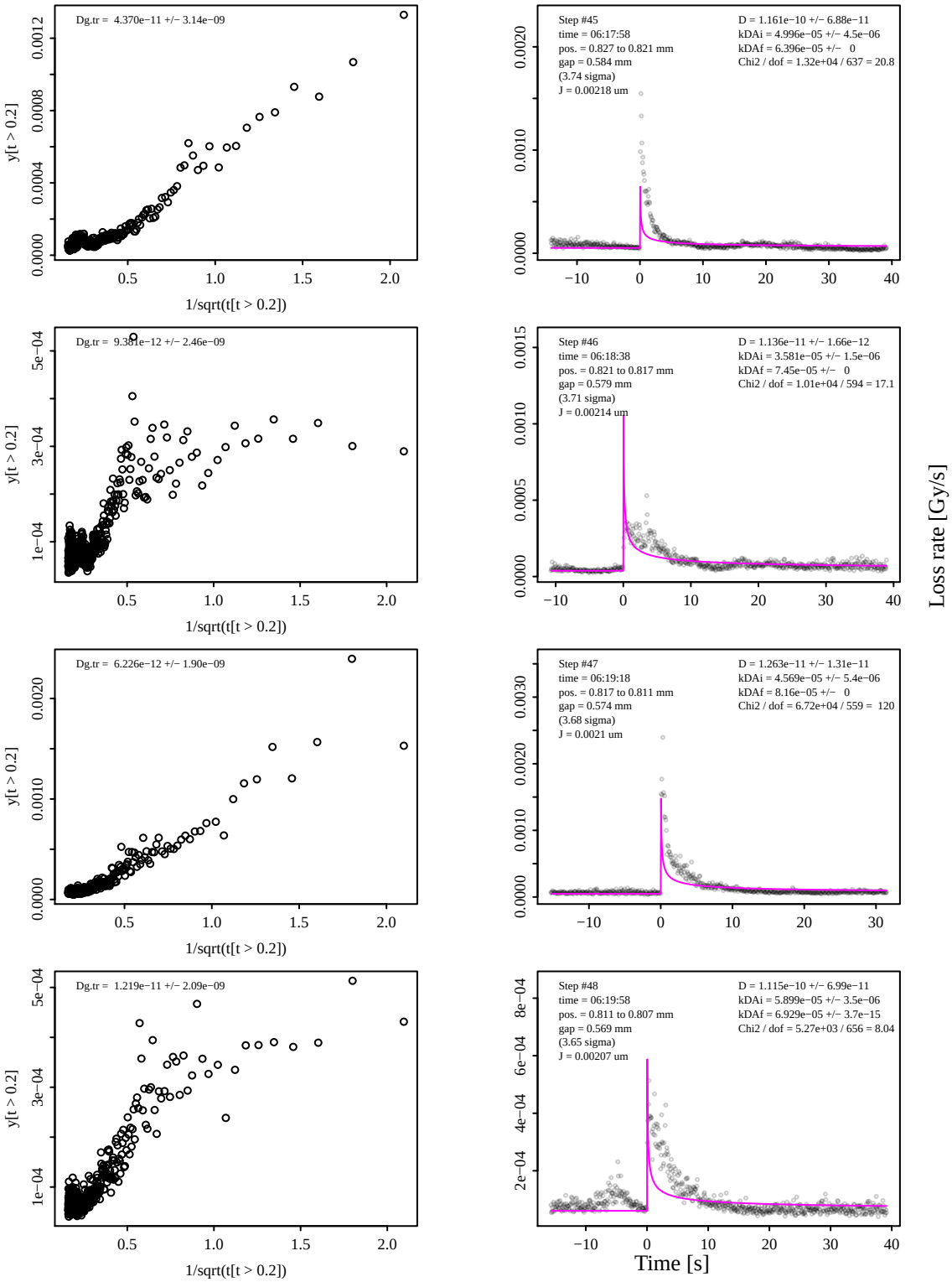

Dg.tr = 5.182e−11 +/− 2.08e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #25
time = 06:05:49
pos. = 0.98 to 0.97 mm
gap = 0.735 mm
(4.71 sigma)
J = 0.00345 um
D = 1.780e−11 +/− 4.37e−12
kDAi = 9.628e−06 +/− 0
kDAf = 3.186e−05 +/− 0
Chi2 / dof = 3.96e+05 / 231 = 1.72e+03
Dg.tr = 2.912e−10 +/− 1.07e−08
Step #26
time = 06:06:04
pos. = 0.97 to 0.96 mm
gap = 0.725 mm
(4.65 sigma)
J = 0.00336 um
D = 5.105e−12 +/− 5.04e−13
kDAi = 8.077e−06 +/− 0
kDAf = 5.407e−05 +/− 0
Chi2 / dof = 6.74e+05 / 221 = 3.05e+03
Dg.tr = 1.394e−11 +/− 4.62e−10
Step #27
time = 06:06:19
pos. = 0.96 to 0.951 mm
gap = 0.716 mm
(4.59 sigma)
J = 0.00327 um
D = 8.310e−11 +/− 1.95e−11
kDAi = 1.71e−05 +/− 0
kDAf = 2.112e−05 +/− 0
Chi2 / dof = 2.66e+05 / 487 = 546
Dg.tr = 7.946e−11 +/− 3.25e−09
Step #28
time = 06:06:55
pos. = 0.951 to 0.941 mm
gap = 0.706 mm
(4.52 sigma)
J = 0.00318 um
D = 2.974e−11 +/− 5.95e−12
kDAi = 1.439e−05 +/− 3.5e−07
kDAf = 4.375e−05 +/− 0
Chi2 / dof = 7.64e+05 / 291 = 2.62e+03
Loss rate [Gy/s]
Time [s]

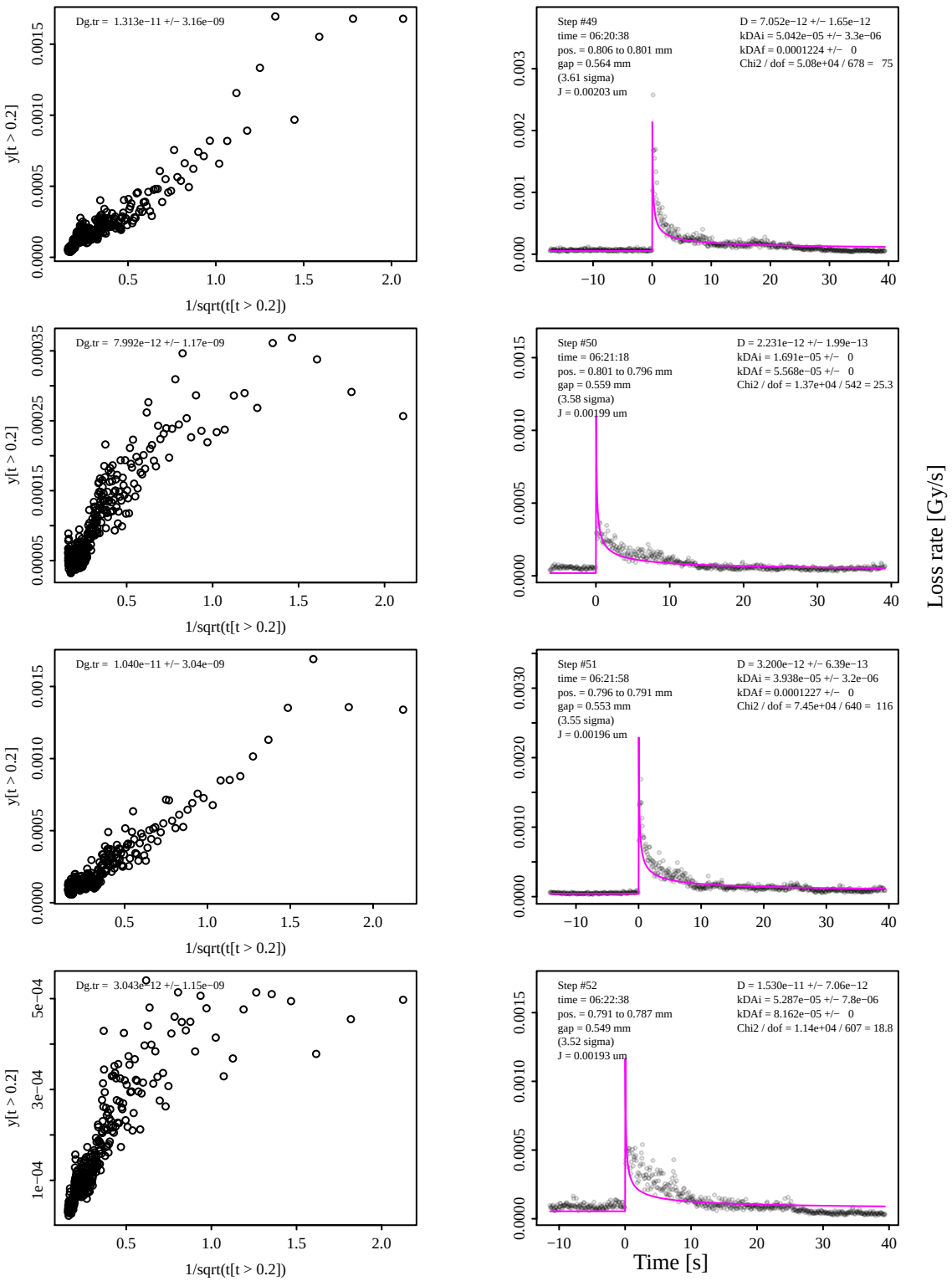

Dg.tr = 1.393e−11 +/− 3.01e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #29
time = 06:07:10
pos. = 0.941 to 0.931 mm
gap = 0.696 mm
(4.46 sigma)
J = 0.00309 um
D = 1.440e−11 +/− 1.24e−12
kDAi = 9.726e−06 +/− 0
kDAf = 2.156e−05 +/− 2.3e−17
Chi2 / dof = 4.66e+04 / 864 = 53.9
Dg.tr = 1.042e−10 +/− 3.94e−09
Step #30
time = 06:08:18
pos. = 0.931 to 0.921 mm
gap = 0.686 mm
(4.4 sigma)
J = 0.00301 um
D = 1.099e−11 +/− 1.99e−12
kDAi = 1.287e−05 +/− 4.6e−07
kDAf = 5.312e−05 +/− 0
Chi2 / dof = 9.31e+04 / 491 = 190
Dg.tr = 1.277e−11 +/− 9.86e−10
Step #31
time = 06:08:43
pos. = 0.921 to 0.912 mm
gap = 0.677 mm
(4.34 sigma)
J = 0.00292 um
D = 1.067e−11 +/− 2.78e−14
kDAi = 1.422e−05 +/− 0
kDAf = 3.42e−05 +/− 0
Chi2 / dof = 1.53e+05 / 384 = 397
Dg.tr = 8.585e−11 +/− 3.66e−09
Step #32
time = 06:09:08
pos. = 0.911 to 0.901 mm
gap = 0.666 mm
(4.27 sigma)
J = 0.00284 um
D = 2.390e−12 +/− 6.68e−13
kDAi = 7.965e−06 +/− 0
kDAf = 6.765e−05 +/− 0
Chi2 / dof = 2.58e+05 / 376 = 686
Loss rate [Gy/s]
Time [s]

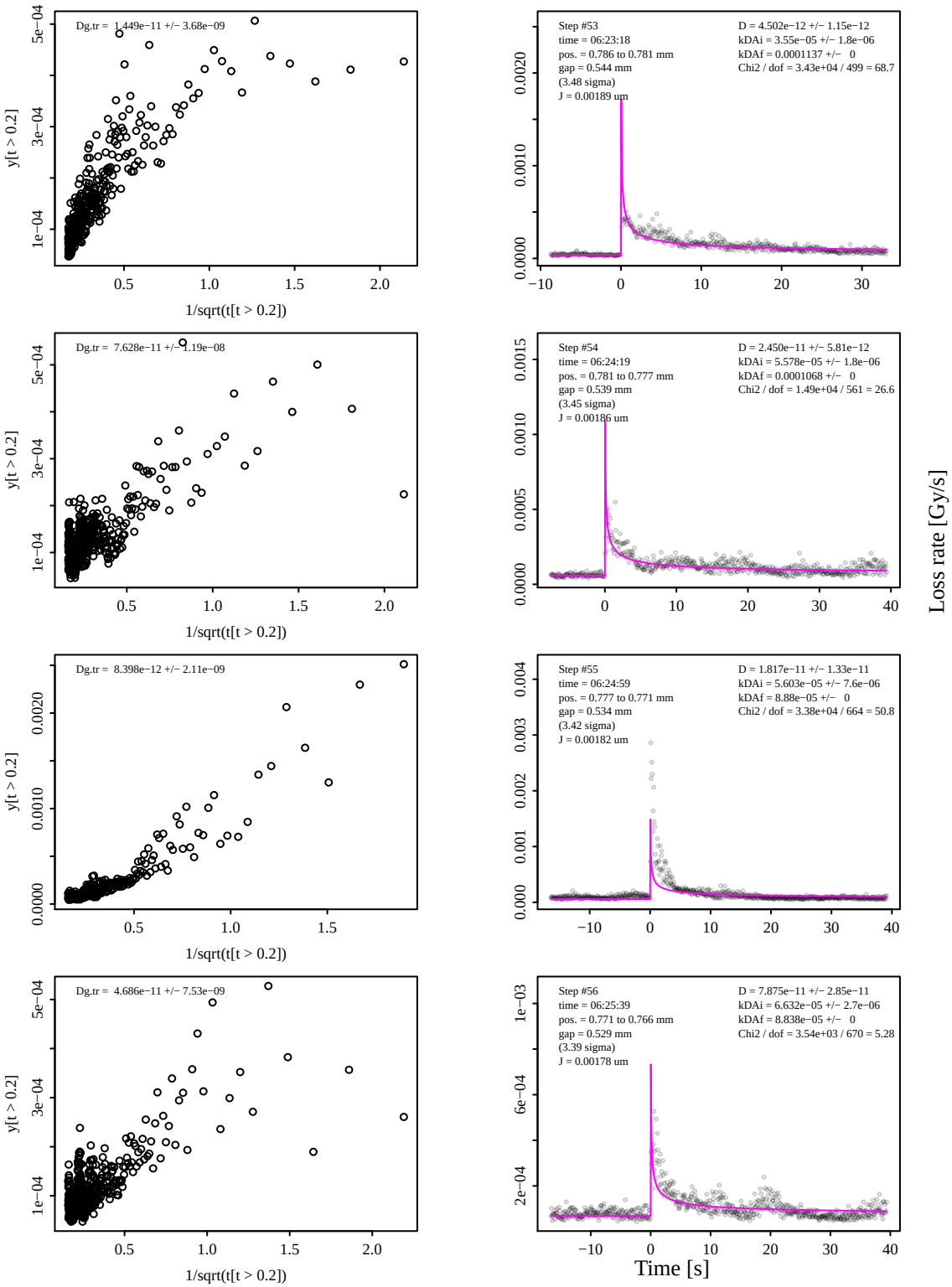

Dg.tr = 5.528e−11 +/− 3.10e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #33
time = 06:09:33
pos. = 0.901 to 0.891 mm
gap = 0.656 mm
(4.21 sigma)
J = 0.00275 um
D = 6.123e−12 +/− 1.29e−12
kDAi = 1.404e−05 +/− 0
kDAf = 6.768e−05 +/− 0
Chi2 / dof = 3.88e+04 / 376 = 103
Dg.tr = 1.901e−11 +/− 6.65e−10
Step #34
time = 06:09:58
pos. = 0.891 to 0.881 mm
gap = 0.646 mm
(4.14 sigma)
J = 0.00267 um
D = 1.075e−11 +/− 9.85e−13
kDAi = 1.528e−05 +/− 0
kDAf = 4.269e−05 +/− 7e−12
Chi2 / dof = 3.22e+05 / 794 = 406
Dg.tr = 2.789e−11 +/− 2.24e−09
Step #35
time = 06:11:15
pos. = 0.881 to 0.876 mm
gap = 0.639 mm
(4.09 sigma)
J = 0.00261 um
D = 3.480e−11 +/− 5.94e−12
kDAi = 2.62e−05 +/− 4.7e−07
kDAf = 4.529e−05 +/− 0
Chi2 / dof = 5.1e+03 / 664 = 7.68
Dg.tr = 2.760e−11 +/− 4.00e−09
Step #36
time = 06:11:40
pos. = 0.876 to 0.871 mm
gap = 0.634 mm
(4.06 sigma)
J = 0.00256 um
D = 3.032e−12 +/− 3.78e−12
kDAi = 1.727e−05 +/− 8.2e−06
kDAf = 6.81e−05 +/− 0
Chi2 / dof = 1.33e+04 / 362 = 36.9
Loss rate [Gy/s]
Time [s]

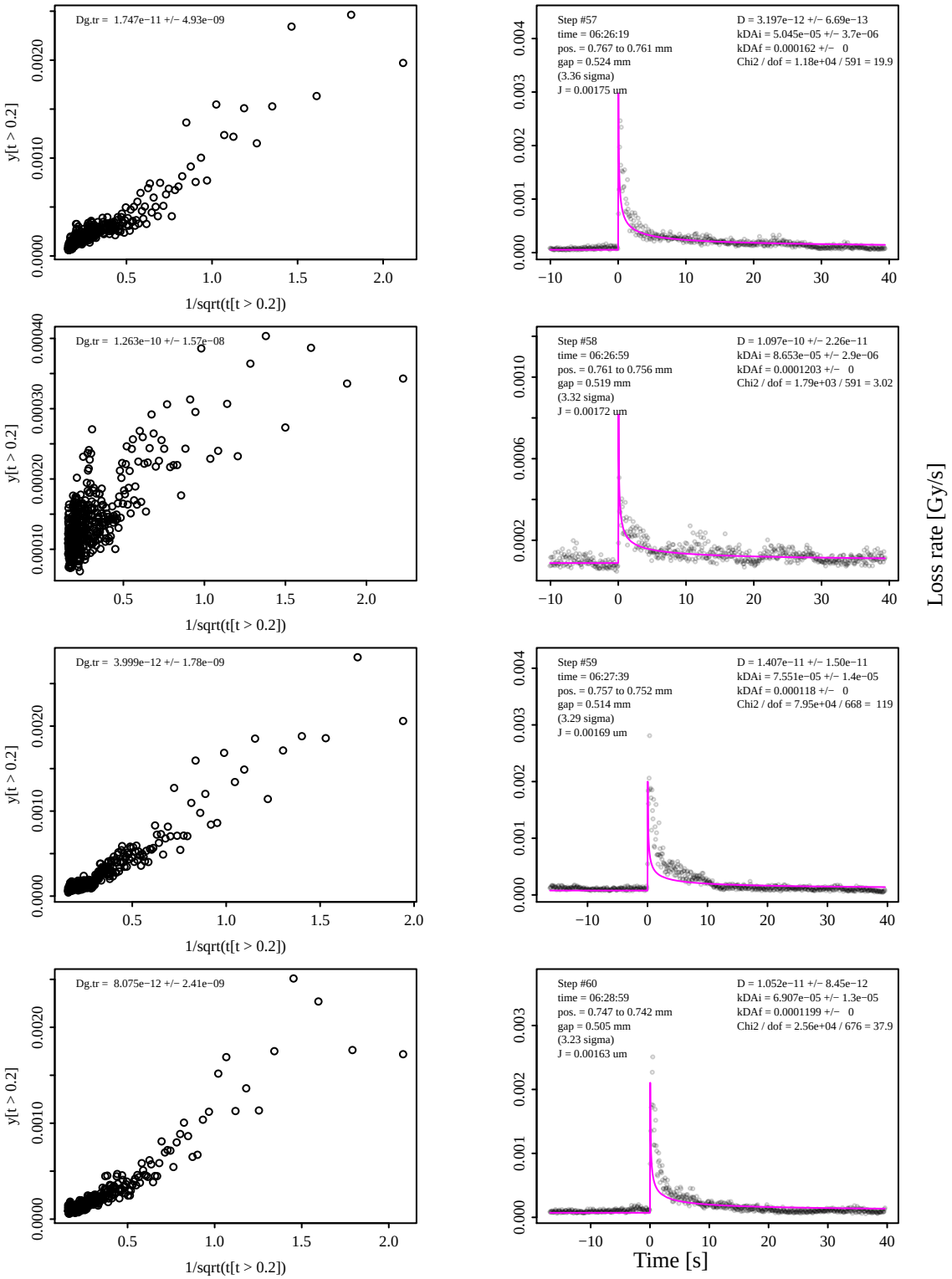

Dg.tr = 1.770e−11 +/− 3.00e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #37
time = 06:12:05
pos. = 0.871 to 0.867 mm
gap = 0.629 mm
(4.03 sigma)
J = 0.00253 um
D = 1.268e−10 +/− 1.87e−10
kDAi = 5.327e−05 +/− 8.5e−06
kDAf = 6.388e−05 +/− 0
Chi2 / dof = 2.9e+04 / 409 = 70.9
Dg.tr = 2.038e−11 +/− 2.82e−09
Step #38
time = 06:12:30
pos. = 0.867 to 0.861 mm
gap = 0.624 mm
(4 sigma)
J = 0.00249 um
D = 2.061e−12 +/− 9.30e−13
kDAi = 1.642e−05 +/− 2.7e−06
kDAf = 7.824e−05 +/− 0
Chi2 / dof = 2.97e+04 / 373 = 79.7
Loss rate [Gy/s]
Dg.tr = 1.556e−11 +/− 2.90e−09
Step #39
time = 06:12:55
pos. = 0.861 to 0.857 mm
gap = 0.619 mm
(3.97 sigma)
J = 0.00245 um
D = 3.161e−11 +/− 3.92e−12
kDAi = 4.58e−05 +/− 7.9e−07
kDAf = 7.008e−05 +/− 0
Chi2 / dof = 8.39e+03 / 373 = 22.5
Dg.tr = 4.166e−11 +/− 5.63e−09
Step #40
time = 06:13:20
pos. = 0.857 to 0.851 mm
gap = 0.614 mm
(3.93 sigma)
J = 0.00241 um
D = 1.235e−11 +/− 5.51e−12
kDAi = 3.91e−05 +/− 4.4e−06
kDAf = 0.0001001 +/− 0
Chi2 / dof = 1.22e+05 / 384 = 318
Time [s]

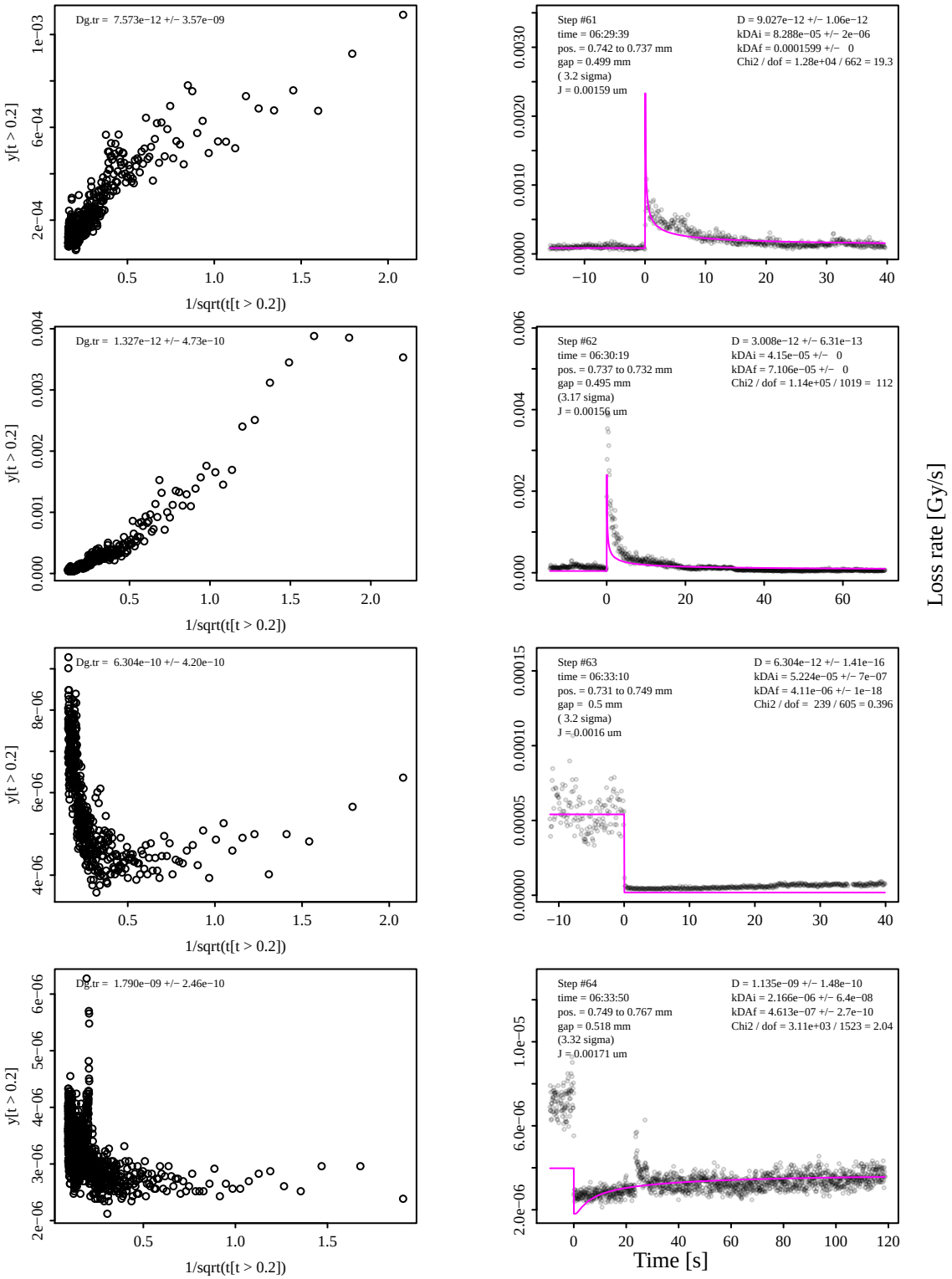

Dg.tr = 3.791e−11 +/− 5.79e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #41
time = 06:13:45
pos. = 0.851 to 0.846 mm
gap = 0.609 mm
(3.9 sigma)
J = 0.00237 um
D = 7.456e−11 +/− 1.31e−11
kDAi = 6.158e−05 +/− 1.2e−06
kDAf = 9.05e−05 +/− 0
Chi2 / dof = 4.73e+03 / 384 = 12.3
Dg.tr = 1.682e−11 +/− 3.23e−09
Step #42
time = 06:14:10
pos. = 0.847 to 0.841 mm
gap = 0.604 mm
(3.87 sigma)
J = 0.00233 um
D = 3.030e−12 +/− 8.83e−13
kDAi = 2.551e−05 +/− 6.9e−08
kDAf = 9.439e−05 +/− 0
Chi2 / dof = 1.47e+04 / 389 = 37.9
Dg.tr = 7.597e−12 +/− 1.60e−09
Step #43
time = 06:15:00
pos. = 0.837 to 0.831 mm
gap = 0.594 mm
(3.81 sigma)
J = 0.00225 um
D = 1.941e−12 +/− 9.63e−14
kDAi = 2.392e−05 +/− 0
kDAf = 8.94e−05 +/− 0
Chi2 / dof = 5.87e+04 / 599 = 97.9
Dg.tr = 2.272e−10 +/− 1.57e−08
Step #44
time = 06:17:18
pos. = 0.832 to 0.827 mm
gap = 0.589 mm
(3.78 sigma)
J = 0.00222 um
D = 3.012e−11 +/− 3.27e−12
kDAi = 4.148e−05 +/− 1e−06
kDAf = 7.894e−05 +/− 0
Chi2 / dof = 1.32e+04 / 569 = 23.1
Loss rate [Gy/s]
Time [s]

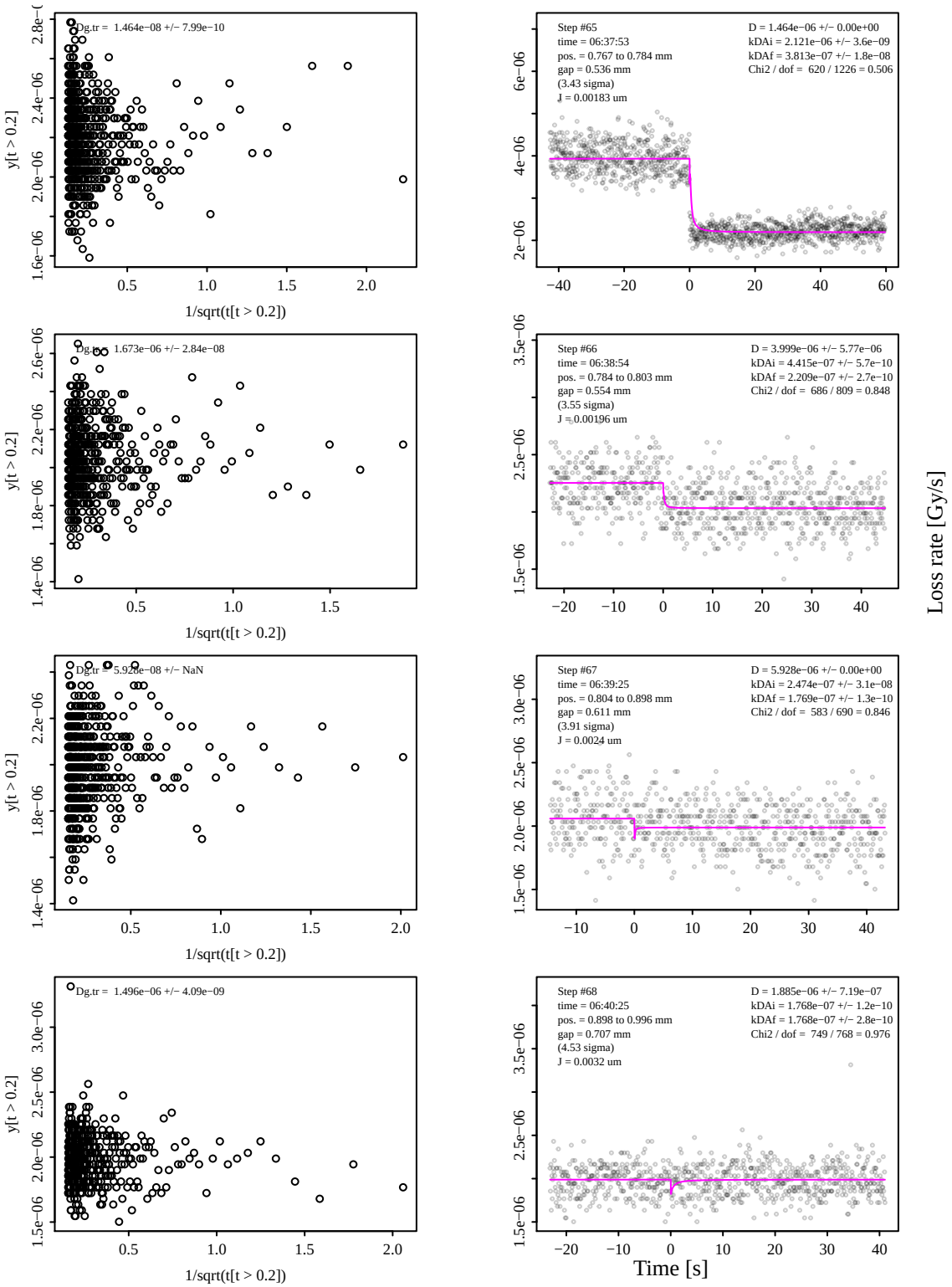

Dg.tr = 4.370e−11 +/− 3.14e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #45
time = 06:17:58
pos. = 0.827 to 0.821 mm
gap = 0.584 mm
(3.74 sigma)
J = 0.00218 um
D = 1.161e−10 +/− 6.88e−11
kDAi = 4.996e−05 +/− 4.5e−06
kDAf = 6.396e−05 +/− 0
Chi2 / dof = 1.32e+04 / 637 = 20.8
Dg.tr = 9.381e−12 +/− 2.46e−09
Step #46
time = 06:18:38
pos. = 0.821 to 0.817 mm
gap = 0.579 mm
(3.71 sigma)
J = 0.00214 um
D = 1.136e−11 +/− 1.66e−12
kDAi = 3.581e−05 +/− 1.5e−06
kDAf = 7.45e−05 +/− 0
Chi2 / dof = 1.01e+04 / 594 = 17.1
Dg.tr = 6.226e−12 +/− 1.90e−09
Step #47
time = 06:19:18
pos. = 0.817 to 0.811 mm
gap = 0.574 mm
(3.68 sigma)
J = 0.0021 um
D = 1.263e−11 +/− 1.31e−11
kDAi = 4.569e−05 +/− 5.4e−06
kDAf = 8.16e−05 +/− 0
Chi2 / dof = 6.72e+04 / 559 = 120
Dg.tr = 1.219e−11 +/− 2.09e−09
Step #48
time = 06:19:58
pos. = 0.811 to 0.807 mm
gap = 0.569 mm
(3.65 sigma)
J = 0.00207 um
D = 1.115e−10 +/− 6.99e−11
kDAi = 5.899e−05 +/− 3.5e−06
kDAf = 6.929e−05 +/− 3.7e−15
Chi2 / dof = 5.27e+03 / 656 = 8.04
Loss rate [Gy/s]
Time [s]

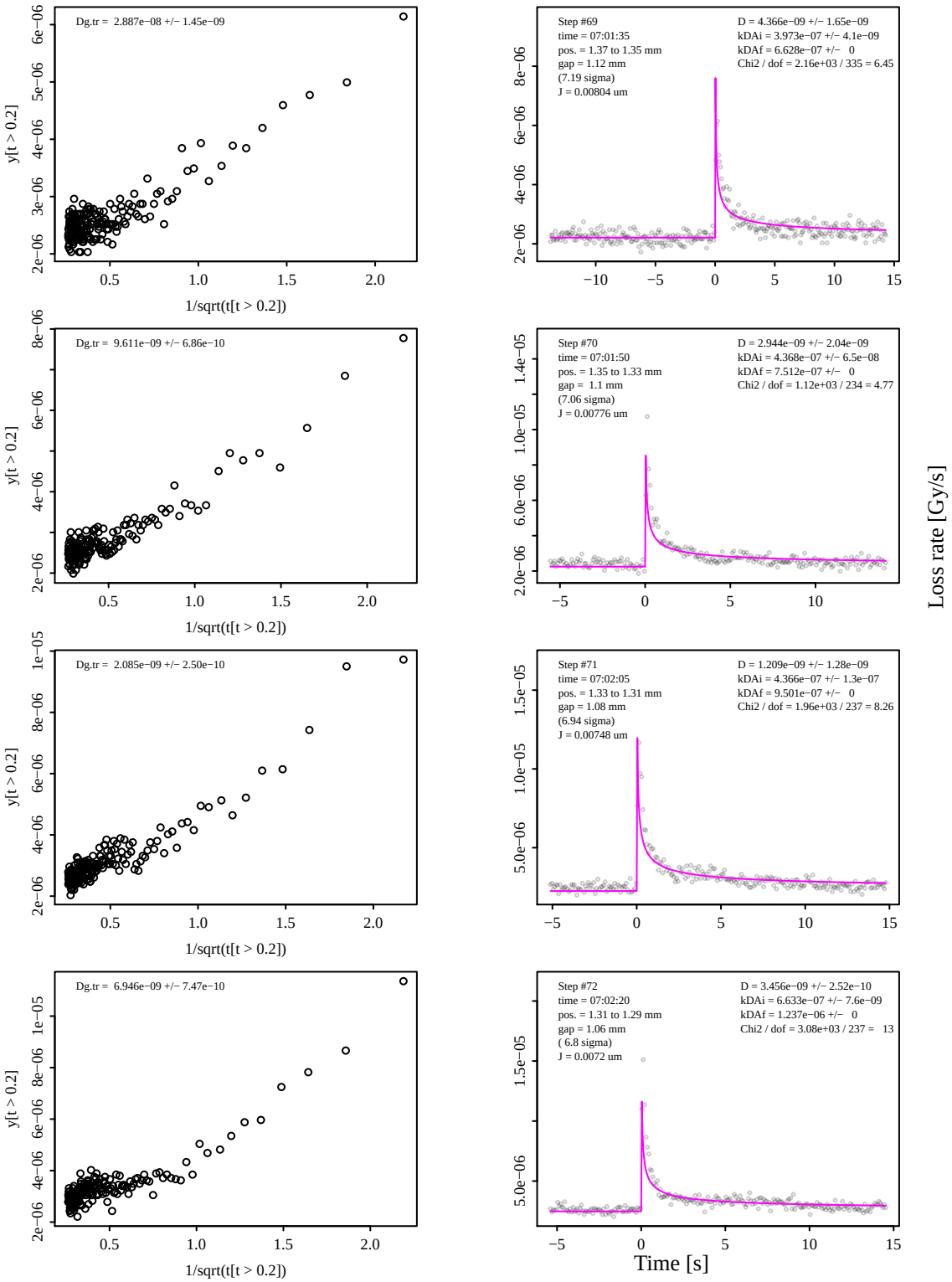

Dg.tr = 1.313e−11 +/− 3.16e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #49
time = 06:20:38
pos. = 0.806 to 0.801 mm
gap = 0.564 mm
(3.61 sigma)
J = 0.00203 um
D = 7.052e−12 +/− 1.65e−12
kDAi = 5.042e−05 +/− 3.3e−06
kDAf = 0.0001224 +/− 0
Chi2 / dof = 5.08e+04 / 678 = 75
Dg.tr = 7.992e−12 +/− 1.17e−09
Step #50
time = 06:21:18
pos. = 0.801 to 0.796 mm
gap = 0.559 mm
(3.58 sigma)
J = 0.00199 um
D = 2.231e−12 +/− 1.99e−13
kDAi = 1.691e−05 +/− 0
kDAf = 5.568e−05 +/− 0
Chi2 / dof = 1.37e+04 / 542 = 25.3
Dg.tr = 1.040e−11 +/− 3.04e−09
Step #51
time = 06:21:58
pos. = 0.796 to 0.791 mm
gap = 0.553 mm
(3.55 sigma)
J = 0.00196 um
D = 3.200e−12 +/− 6.39e−13
kDAi = 3.938e−05 +/− 3.2e−06
kDAf = 0.0001227 +/− 0
Chi2 / dof = 7.45e+04 / 640 = 116
Dg.tr = 3.043e−12 +/− 1.15e−09
Step #52
time = 06:22:38
pos. = 0.791 to 0.787 mm
gap = 0.549 mm
(3.52 sigma)
J = 0.00193 um
D = 1.530e−11 +/− 7.06e−12
kDAi = 5.287e−05 +/− 7.8e−06
kDAf = 8.162e−05 +/− 0
Chi2 / dof = 1.14e+04 / 607 = 18.8
Loss rate [Gy/s]
Time [s]

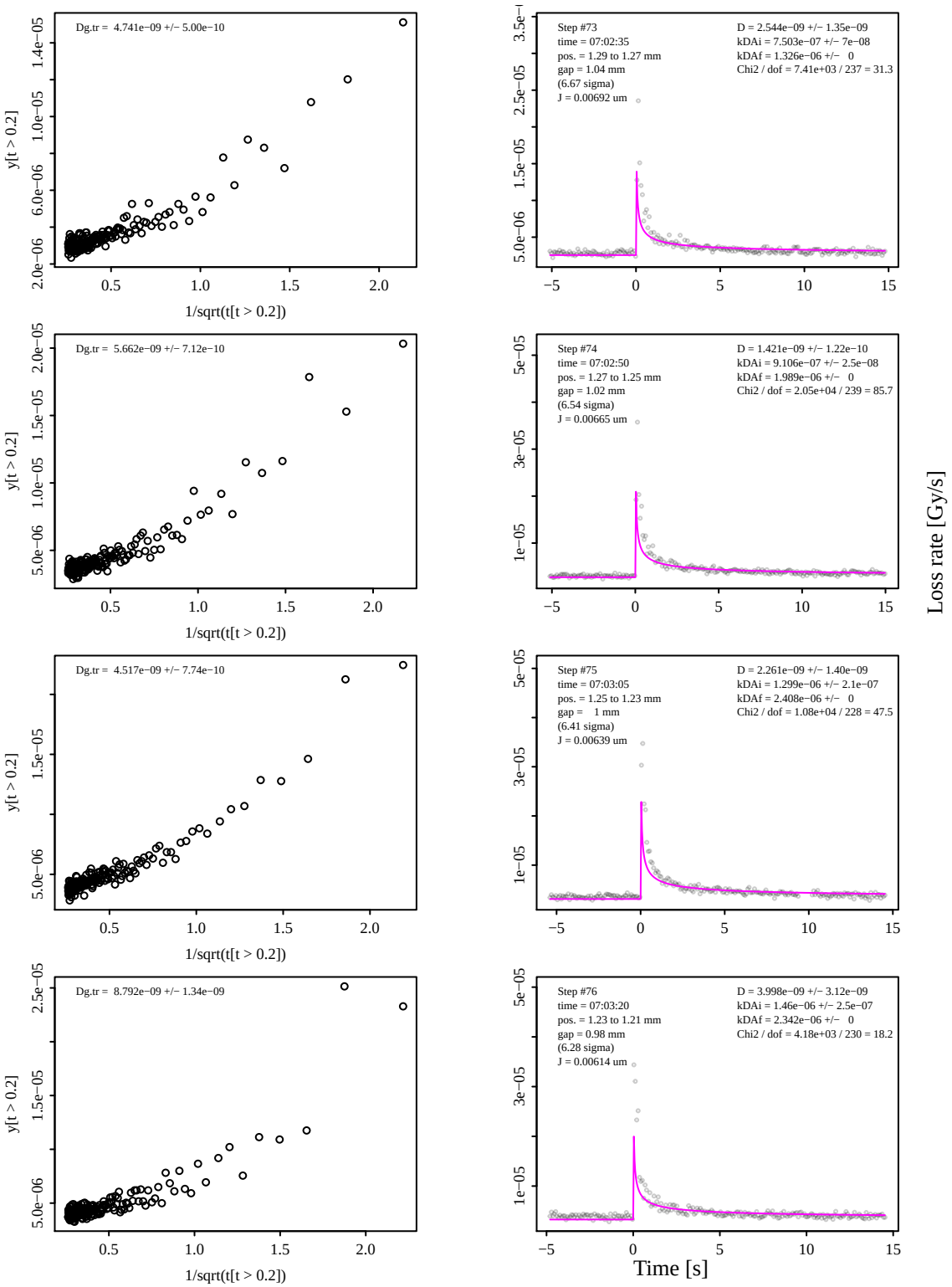

Dg.tr = 1.449e−11 +/− 3.68e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #53
time = 06:23:18
pos. = 0.786 to 0.781 mm
gap = 0.544 mm
(3.48 sigma)
J = 0.00189 um
D = 4.502e−12 +/− 1.15e−12
kDAi = 3.55e−05 +/− 1.8e−06
kDAf = 0.0001137 +/− 0
Chi2 / dof = 3.43e+04 / 499 = 68.7
Dg.tr = 7.628e−11 +/− 1.19e−08
Step #54
time = 06:24:19
pos. = 0.781 to 0.777 mm
gap = 0.539 mm
(3.45 sigma)
J = 0.00186 um
D = 2.450e−11 +/− 5.81e−12
kDAi = 5.578e−05 +/− 1.8e−06
kDAf = 0.0001068 +/− 0
Chi2 / dof = 1.49e+04 / 561 = 26.6
Dg.tr = 8.398e−12 +/− 2.11e−09
Step #55
time = 06:24:59
pos. = 0.777 to 0.771 mm
gap = 0.534 mm
(3.42 sigma)
J = 0.00182 um
D = 1.817e−11 +/− 1.33e−11
kDAi = 5.603e−05 +/− 7.6e−06
kDAf = 8.88e−05 +/− 0
Chi2 / dof = 3.38e+04 / 664 = 50.8
Dg.tr = 4.686e−11 +/− 7.53e−09
Step #56
time = 06:25:39
pos. = 0.771 to 0.766 mm
gap = 0.529 mm
(3.39 sigma)
J = 0.00178 um
D = 7.875e−11 +/− 2.85e−11
kDAi = 6.632e−05 +/− 2.7e−06
kDAf = 8.838e−05 +/− 0
Chi2 / dof = 3.54e+03 / 670 = 5.28
Loss rate [Gy/s]
Time [s]

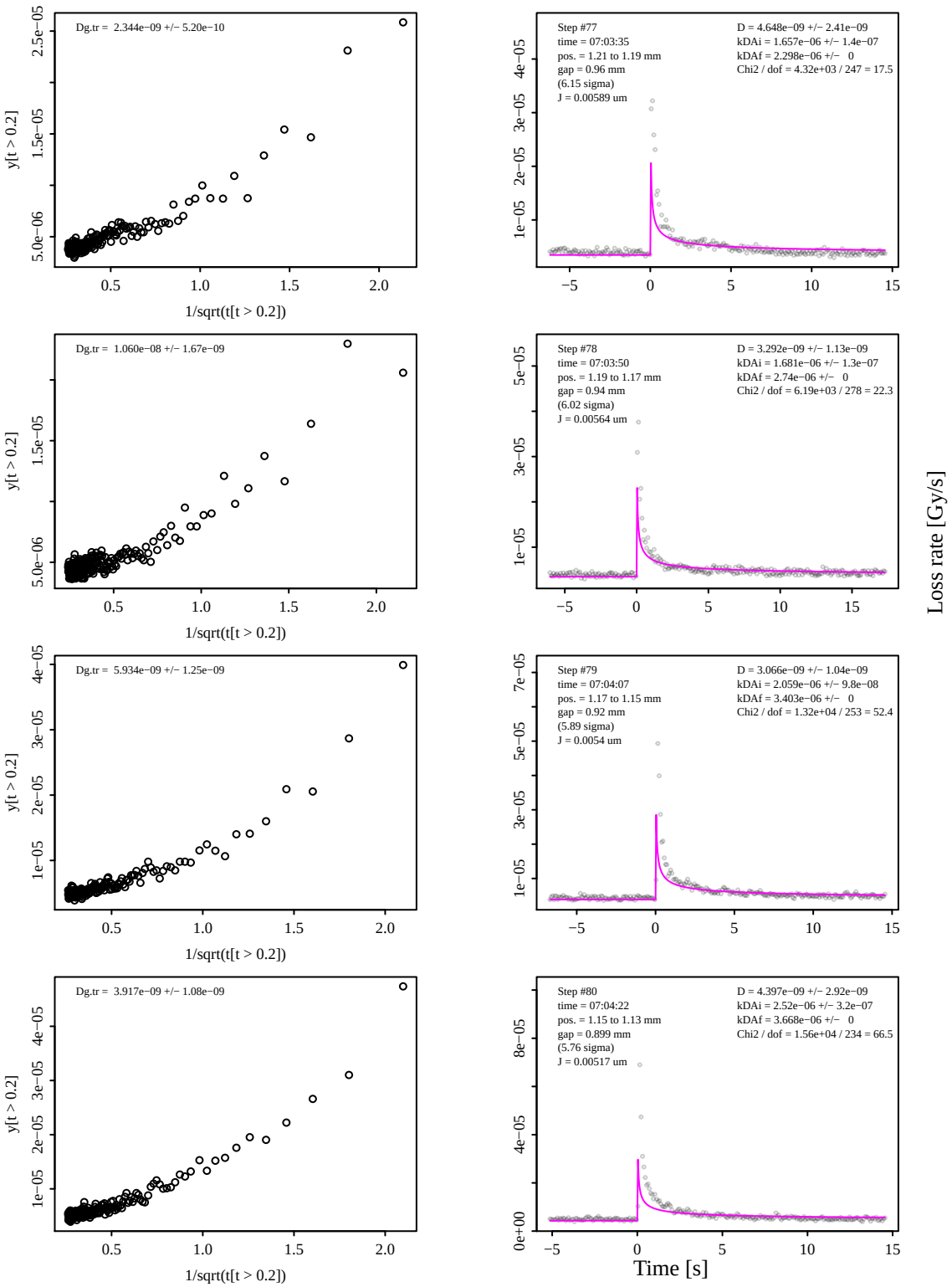

Dg.tr = 1.747e−11 +/− 4.93e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #57
time = 06:26:19
pos. = 0.767 to 0.761 mm
gap = 0.524 mm
(3.36 sigma)
J = 0.00175 um
D = 3.197e−12 +/− 6.69e−13
kDAi = 5.045e−05 +/− 3.7e−06
kDAf = 0.000162 +/− 0
Chi2 / dof = 1.18e+04 / 591 = 19.9
Dg.tr = 1.263e−10 +/− 1.57e−08
Step #58
time = 06:26:59
pos. = 0.761 to 0.756 mm
gap = 0.519 mm
(3.32 sigma)
J = 0.00172 um
D = 1.097e−10 +/− 2.26e−11
kDAi = 8.653e−05 +/− 2.9e−06
kDAf = 0.0001203 +/− 0
Chi2 / dof = 1.79e+03 / 591 = 3.02
Dg.tr = 3.999e−12 +/− 1.78e−09
Step #59
time = 06:27:39
pos. = 0.757 to 0.752 mm
gap = 0.514 mm
(3.29 sigma)
J = 0.00169 um
D = 1.407e−11 +/− 1.50e−11
kDAi = 7.551e−05 +/− 1.4e−05
kDAf = 0.000118 +/− 0
Chi2 / dof = 7.95e+04 / 668 = 119
Dg.tr = 8.075e−12 +/− 2.41e−09
Step #60
time = 06:28:59
pos. = 0.747 to 0.742 mm
gap = 0.505 mm
(3.23 sigma)
J = 0.00163 um
D = 1.052e−11 +/− 8.45e−12
kDAi = 6.907e−05 +/− 1.3e−05
kDAf = 0.0001199 +/− 0
Chi2 / dof = 2.56e+04 / 676 = 37.9
Loss rate [Gy/s]
Time [s]

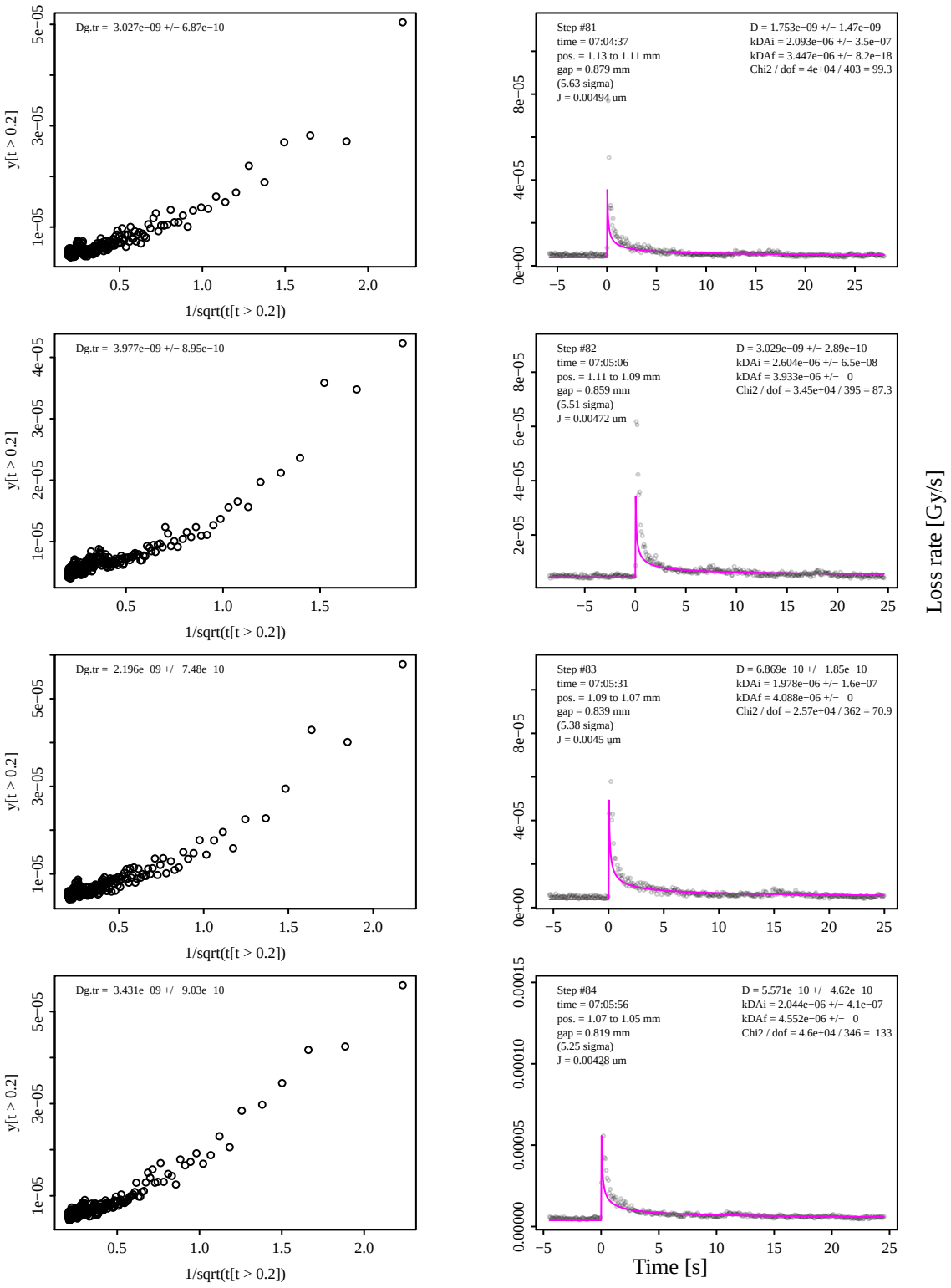

Dg.tr = 7.573e−12 +/− 3.57e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #61
time = 06:29:39
pos. = 0.742 to 0.737 mm
gap = 0.499 mm
(3.2 sigma)
J = 0.00159 um
D = 9.027e−12 +/− 1.06e−12
kDAi = 8.288e−05 +/− 2e−06
kDAf = 0.0001599 +/− 0
Chi2 / dof = 1.28e+04 / 662 = 19.3
Dg.tr = 1.327e−12 +/− 4.73e−10
Step #62
time = 06:30:19
pos. = 0.737 to 0.732 mm
gap = 0.495 mm
(3.17 sigma)
J = 0.00156 um
D = 3.008e−12 +/− 6.31e−13
kDAi = 4.15e−05 +/− 0
kDAf = 7.106e−05 +/− 0
Chi2 / dof = 1.14e+05 / 1019 = 112
Dg.tr = 6.304e−10 +/− 4.20e−10
Step #63
time = 06:33:10
pos. = 0.731 to 0.749 mm
gap = 0.5 mm
(3.2 sigma)
J = 0.0016 um
D = 6.304e−12 +/− 1.41e−16
kDAi = 5.224e−05 +/− 7e−07
kDAf = 4.11e−06 +/− 1e−18
Chi2 / dof = 239 / 605 = 0.396
Dg.tr = 1.790e−09 +/− 2.46e−10
Step #64
time = 06:33:50
pos. = 0.749 to 0.767 mm
gap = 0.518 mm
(3.32 sigma)
J = 0.00171 um
D = 1.135e−09 +/− 1.48e−10
kDAi = 2.166e−06 +/− 6.4e−08
kDAf = 4.613e−07 +/− 2.7e−10
Chi2 / dof = 3.11e+03 / 1523 = 2.04
Loss rate [Gy/s]
Time [s]

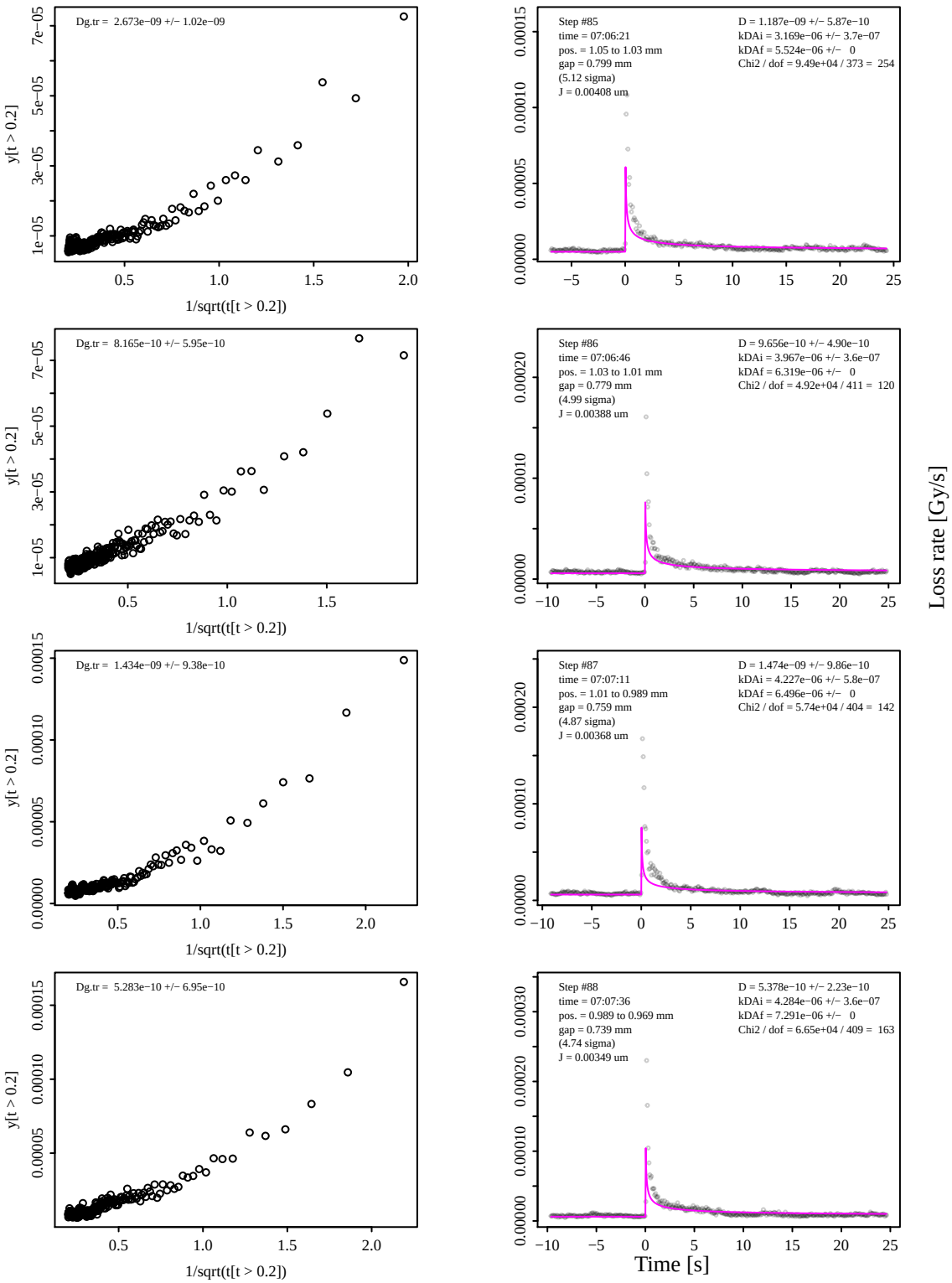

Dg.tr = 1.464e−08 +/− 7.99e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #65
time = 06:37:53
pos. = 0.767 to 0.784 mm
gap = 0.536 mm
(3.43 sigma)
J = 0.00183 um
D = 1.464e−06 +/− 0.00e+00
kDAi = 2.121e−06 +/− 3.6e−09
kDAf = 3.813e−07 +/− 1.8e−08
Chi2 / dof = 620 / 1226 = 0.506
Dg.tr = 1.673e−06 +/− 2.84e−08
Step #66
time = 06:38:54
pos. = 0.784 to 0.803 mm
gap = 0.554 mm
(3.55 sigma)
J = 0.00196 um
D = 3.999e−06 +/− 5.77e−06
kDAi = 4.415e−07 +/− 5.7e−10
kDAf = 2.209e−07 +/− 2.7e−10
Chi2 / dof = 686 / 809 = 0.848
Dg.tr = 5.928e−08 +/− NaN
Step #67
time = 06:39:25
pos. = 0.804 to 0.898 mm
gap = 0.611 mm
(3.91 sigma)
J = 0.0024 um
D = 5.928e−06 +/− 0.00e+00
kDAi = 2.474e−07 +/− 3.1e−08
kDAf = 1.769e−07 +/− 1.3e−10
Chi2 / dof = 583 / 690 = 0.846
Dg.tr = 1.496e−06 +/− 4.09e−09
Step #68
time = 06:40:25
pos. = 0.898 to 0.996 mm
gap = 0.707 mm
(4.53 sigma)
J = 0.0032 um
D = 1.885e−06 +/− 7.19e−07
kDAi = 1.768e−07 +/− 1.2e−10
kDAf = 1.768e−07 +/− 2.8e−10
Chi2 / dof = 749 / 768 = 0.976
Loss rate [Gy/s]
Time [s]

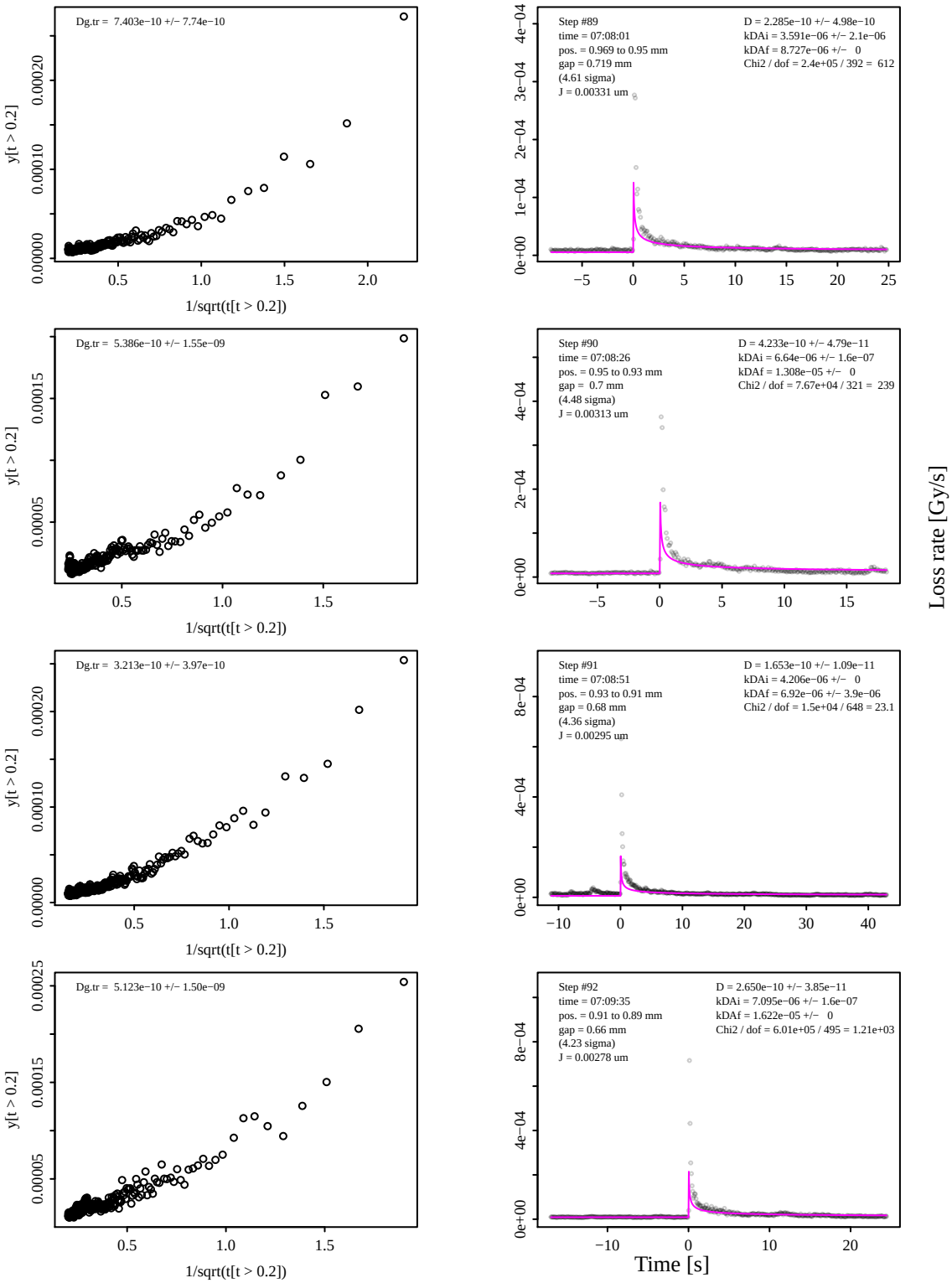

Dg.tr = 2.887e−08 +/− 1.45e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #69
time = 07:01:35
pos. = 1.37 to 1.35 mm
gap = 1.12 mm
(7.19 sigma)
J = 0.00804 um
D = 4.366e−09 +/− 1.65e−09
kDAi = 3.973e−07 +/− 4.1e−09
kDAf = 6.628e−07 +/− 0
Chi2 / dof = 2.16e+03 / 335 = 6.45
Dg.tr = 9.611e−09 +/− 6.86e−10
Step #70
time = 07:01:50
pos. = 1.35 to 1.33 mm
gap = 1.1 mm
(7.06 sigma)
J = 0.00776 um
D = 2.944e−09 +/− 2.04e−09
kDAi = 4.368e−07 +/− 6.5e−08
kDAf = 7.512e−07 +/− 0
Chi2 / dof = 1.12e+03 / 234 = 4.77
Dg.tr = 2.085e−09 +/− 2.50e−10
Step #71
time = 07:02:05
pos. = 1.33 to 1.31 mm
gap = 1.08 mm
(6.94 sigma)
J = 0.00748 um
D = 1.209e−09 +/− 1.28e−09
kDAi = 4.366e−07 +/− 1.3e−07
kDAf = 9.501e−07 +/− 0
Chi2 / dof = 1.96e+03 / 237 = 8.26
Dg.tr = 6.946e−09 +/− 7.47e−10
Step #72
time = 07:02:20
pos. = 1.31 to 1.29 mm
gap = 1.06 mm
(6.8 sigma)
J = 0.0072 um
D = 3.456e−09 +/− 2.52e−10
kDAi = 6.633e−07 +/− 7.6e−09
kDAf = 1.237e−06 +/− 0
Chi2 / dof = 3.08e+03 / 237 = 13
Loss rate [Gy/s]
Time [s]

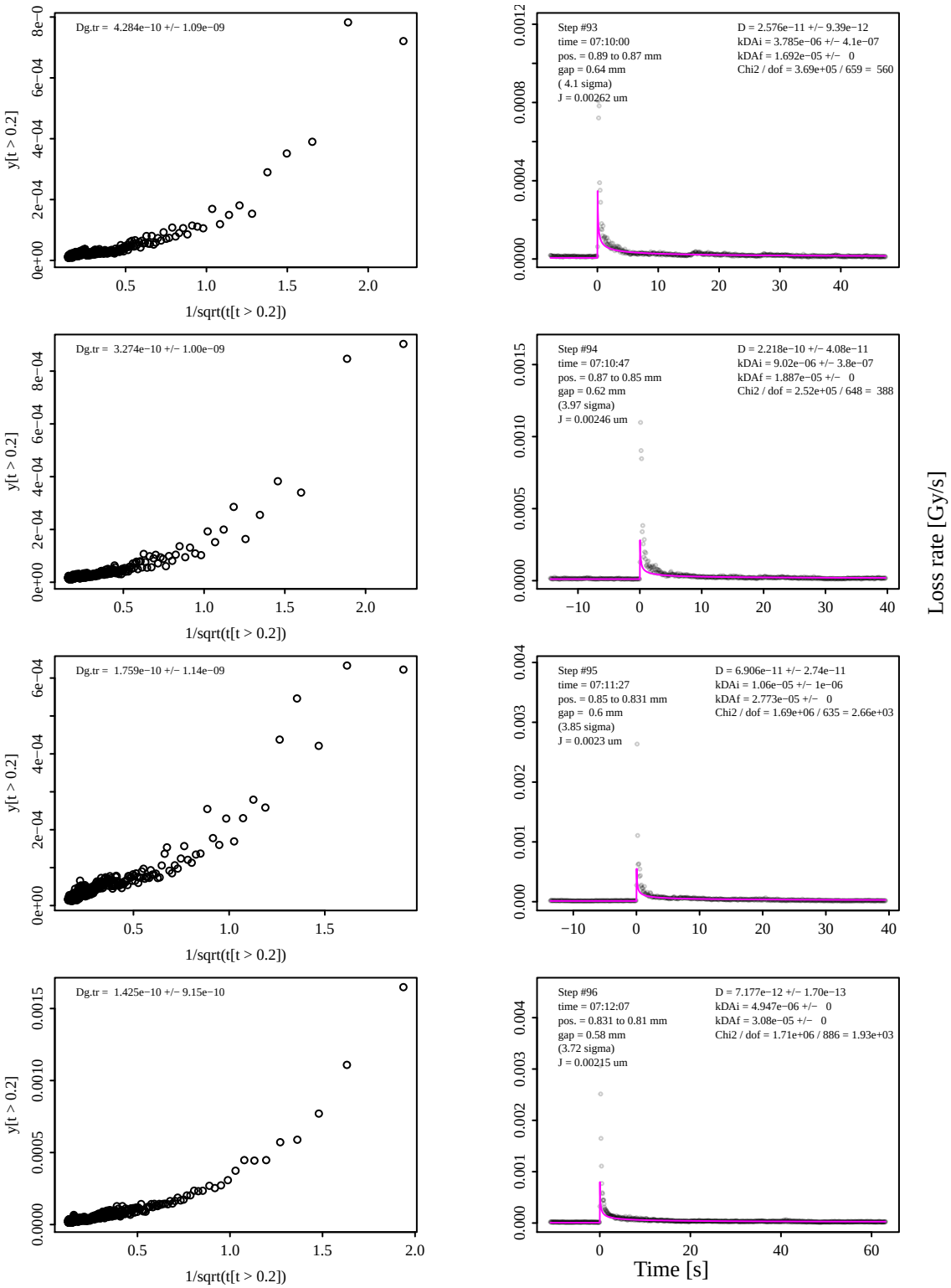

Dg.tr = 4.741e−09 +/− 5.00e−10
Step #73
time = 07:02:35
pos. = 1.29 to 1.27 mm
gap = 1.04 mm
(6.67 sigma)
J = 0.00692 um
D = 2.544e−09 +/− 1.35e−09
kDAi = 7.503e−07 +/− 7e−08
kDAf = 1.326e−06 +/− 0
Chi2 / dof = 7.41e+03 / 237 = 31.3
Dg.tr = 5.662e−09 +/− 7.12e−10
Step #74
time = 07:02:50
pos. = 1.27 to 1.25 mm
gap = 1.02 mm
(6.54 sigma)
J = 0.00665 um
D = 1.421e−09 +/− 1.22e−10
kDAi = 9.106e−07 +/− 2.5e−08
kDAf = 1.989e−06 +/− 0
Chi2 / dof = 2.05e+04 / 239 = 85.7
Dg.tr = 4.517e−09 +/− 7.74e−10
Step #75
time = 07:03:05
pos. = 1.25 to 1.23 mm
gap = 1 mm
(6.41 sigma)
J = 0.00639 um
D = 2.261e−09 +/− 1.40e−09
kDAi = 1.299e−06 +/− 2.1e−07
kDAf = 2.408e−06 +/− 0
Chi2 / dof = 1.08e+04 / 228 = 47.5
Dg.tr = 8.792e−09 +/− 1.34e−09
Step #76
time = 07:03:20
pos. = 1.23 to 1.21 mm
gap = 0.98 mm
(6.28 sigma)
J = 0.00614 um
D = 3.998e−09 +/− 3.12e−09
kDAi = 1.46e−06 +/− 2.5e−07
kDAf = 2.342e−06 +/− 0
Chi2 / dof = 4.18e+03 / 230 = 18.2
y[t > 0.2]
1/sqrt(t[t > 0.2])
Loss rate [Gy/s]
Time [s]

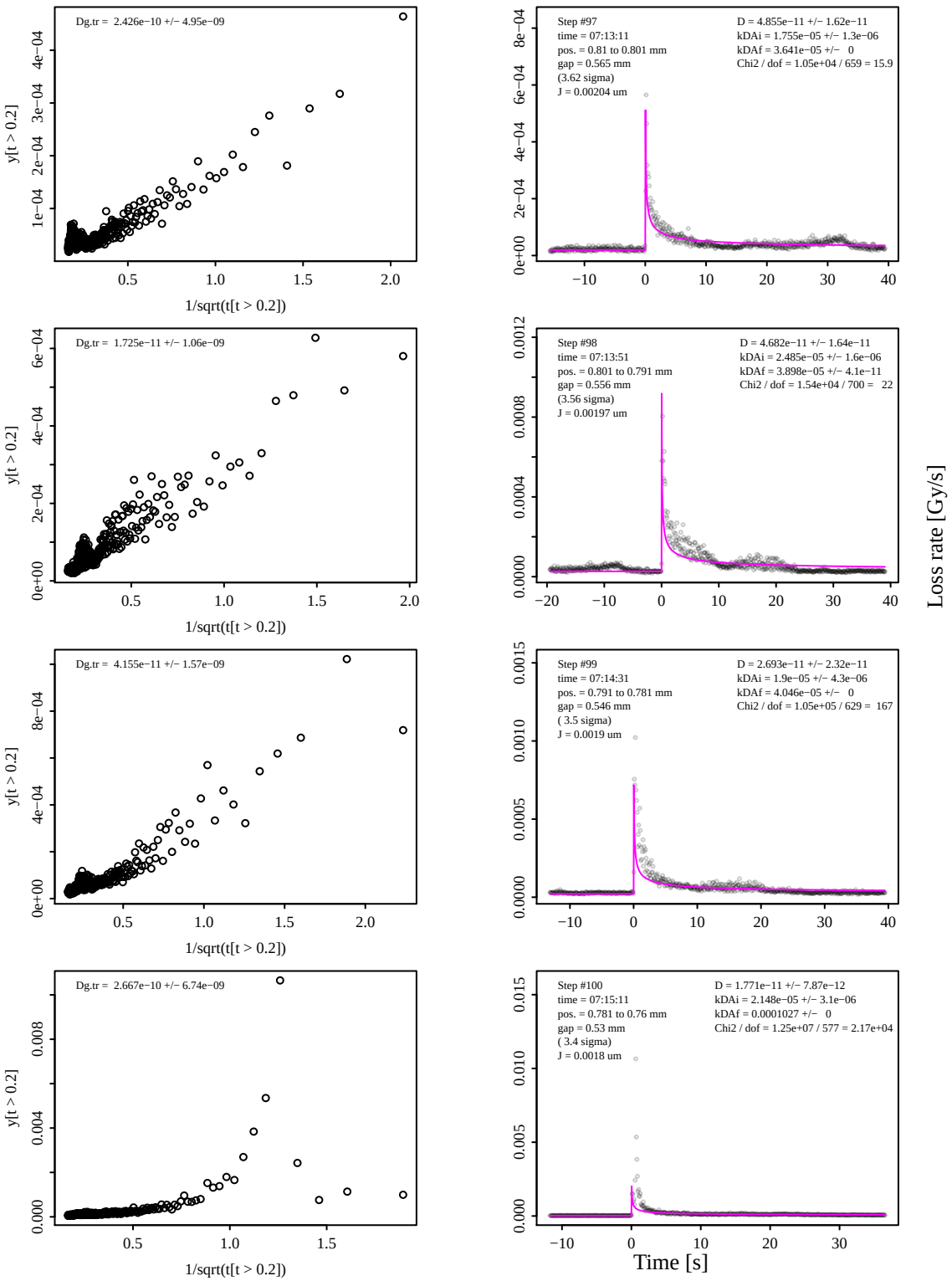

Dg.tr = 2.344e−09 +/− 5.20e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #77
time = 07:03:35
pos. = 1.21 to 1.19 mm
gap = 0.96 mm
(6.15 sigma)
J = 0.00589 um
D = 4.648e−09 +/− 2.41e−09
kDAi = 1.657e−06 +/− 1.4e−07
kDAf = 2.298e−06 +/− 0
Chi2 / dof = 4.32e+03 / 247 = 17.5
Dg.tr = 1.060e−08 +/− 1.67e−09
Step #78
time = 07:03:50
pos. = 1.19 to 1.17 mm
gap = 0.94 mm
(6.02 sigma)
J = 0.00564 um
D = 3.292e−09 +/− 1.13e−09
kDAi = 1.681e−06 +/− 1.3e−07
kDAf = 2.74e−06 +/− 0
Chi2 / dof = 6.19e+03 / 278 = 22.3
Dg.tr = 5.934e−09 +/− 1.25e−09
Step #79
time = 07:04:07
pos. = 1.17 to 1.15 mm
gap = 0.92 mm
(5.89 sigma)
J = 0.0054 um
D = 3.066e−09 +/− 1.04e−09
kDAi = 2.059e−06 +/− 9.8e−08
kDAf = 3.403e−06 +/− 0
Chi2 / dof = 1.32e+04 / 253 = 52.4
Dg.tr = 3.917e−09 +/− 1.08e−09
Step #80
time = 07:04:22
pos. = 1.15 to 1.13 mm
gap = 0.899 mm
(5.76 sigma)
J = 0.00517 um
D = 4.397e−09 +/− 2.92e−09
kDAi = 2.52e−06 +/− 3.2e−07
kDAf = 3.668e−06 +/− 0
Chi2 / dof = 1.56e+04 / 234 = 66.5
Loss rate [Gy/s]
Time [s]

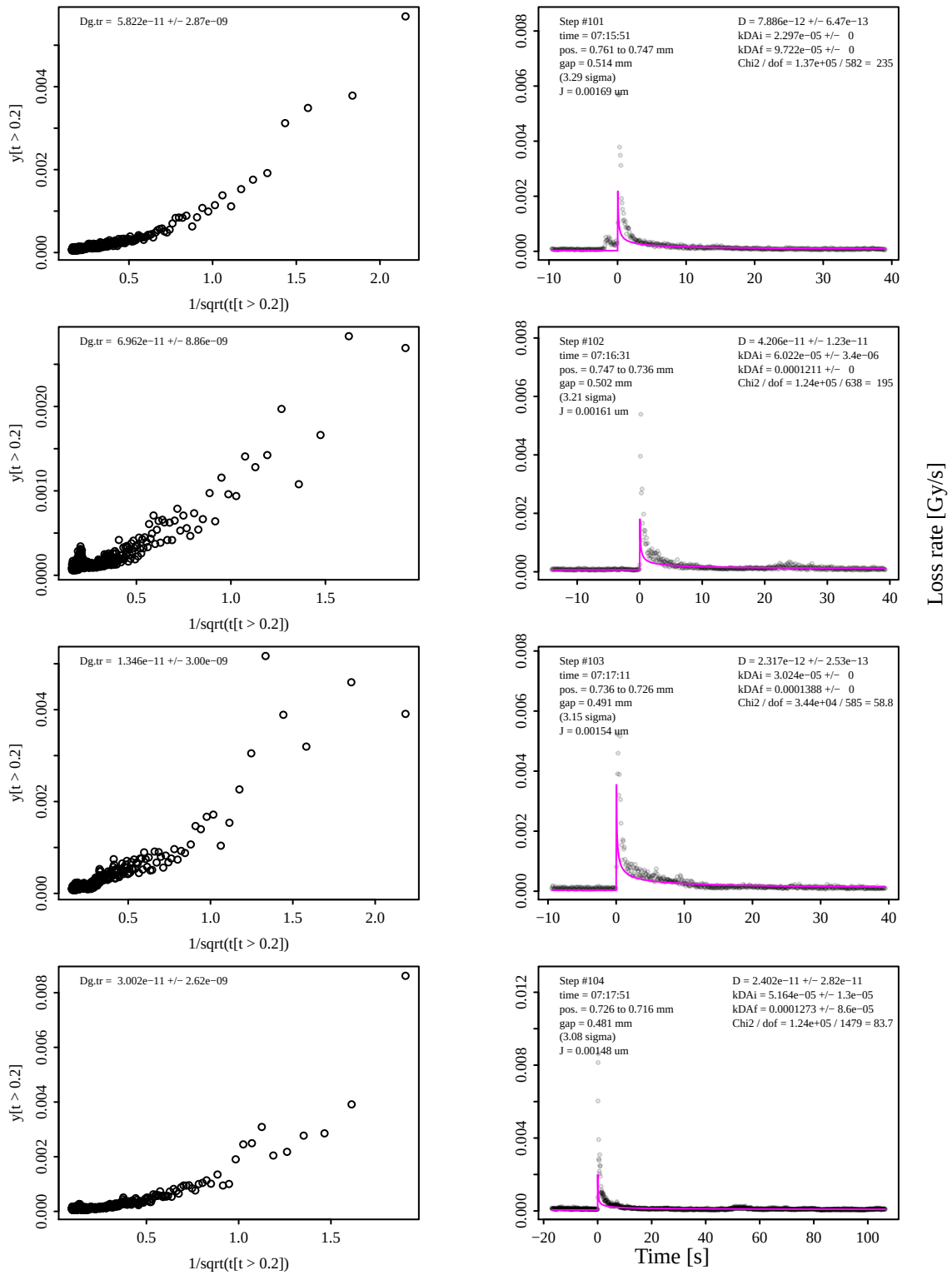

Dg.tr = 3.027e−09 +/− 6.87e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #81
time = 07:04:37
pos. = 1.13 to 1.11 mm
gap = 0.879 mm
(5.63 sigma)
J = 0.00494 um
D = 1.753e−09 +/− 1.47e−09
kDAi = 2.093e−06 +/− 3.5e−07
kDAf = 3.447e−06 +/− 8.2e−18
Chi2 / dof = 4e+04 / 403 = 99.3
Dg.tr = 3.977e−09 +/− 8.95e−10
Step #82
time = 07:05:06
pos. = 1.11 to 1.09 mm
gap = 0.859 mm
(5.51 sigma)
J = 0.00472 um
D = 3.029e−09 +/− 2.89e−10
kDAi = 2.604e−06 +/− 6.5e−08
kDAf = 3.933e−06 +/− 0
Chi2 / dof = 3.45e+04 / 395 = 87.3
Dg.tr = 2.196e−09 +/− 7.48e−10
Step #83
time = 07:05:31
pos. = 1.09 to 1.07 mm
gap = 0.839 mm
(5.38 sigma)
J = 0.0045 um
D = 6.869e−10 +/− 1.85e−10
kDAi = 1.978e−06 +/− 1.6e−07
kDAf = 4.088e−06 +/− 0
Chi2 / dof = 2.57e+04 / 362 = 70.9
Dg.tr = 3.431e−09 +/− 9.03e−10
Step #84
time = 07:05:56
pos. = 1.07 to 1.05 mm
gap = 0.819 mm
(5.25 sigma)
J = 0.00428 um
D = 5.571e−10 +/− 4.62e−10
kDAi = 2.044e−06 +/− 4.1e−07
kDAf = 4.552e−06 +/− 0
Chi2 / dof = 4.6e+04 / 346 = 133
Loss rate [Gy/s]
Time [s]

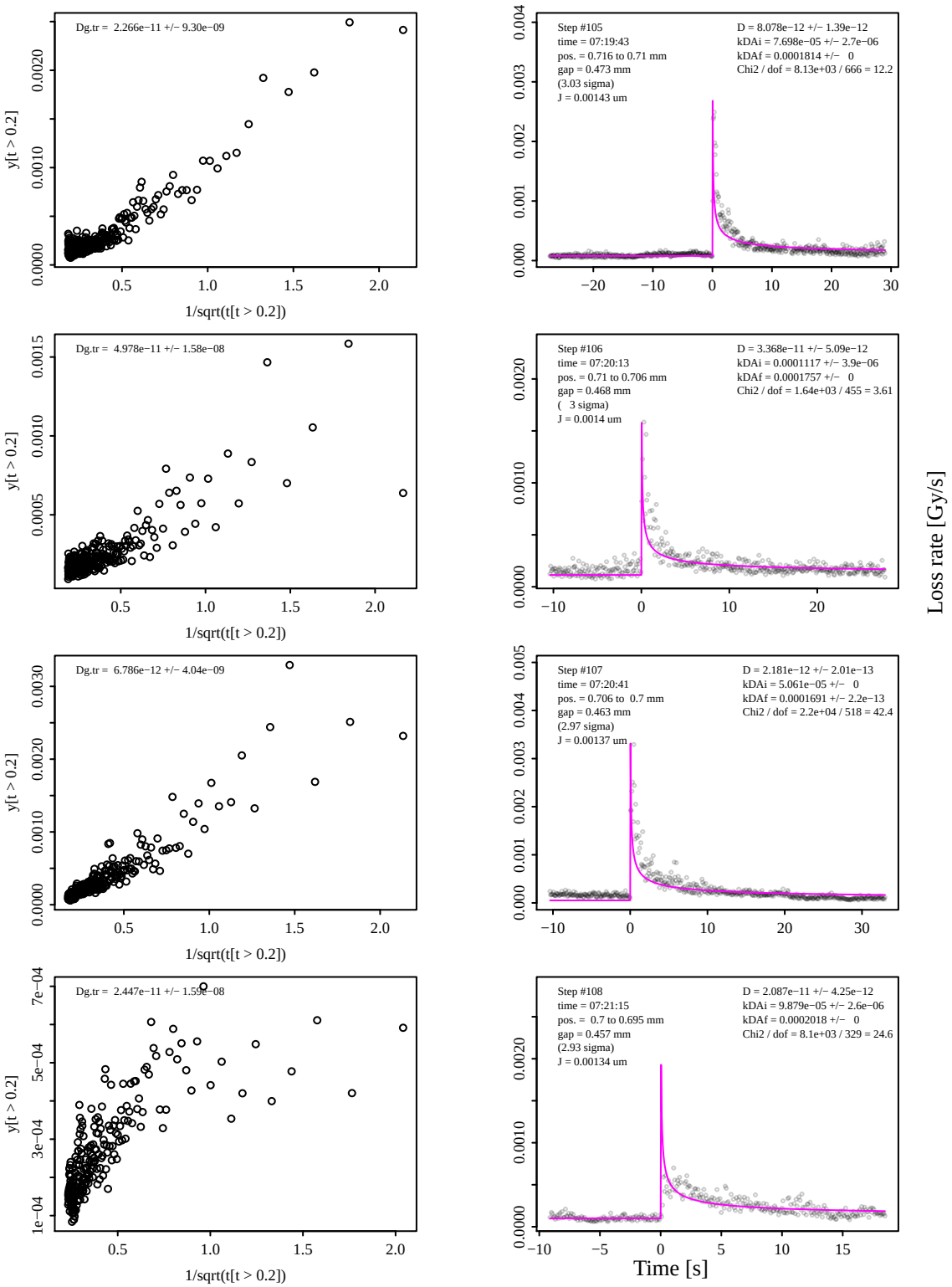

Dg.tr = 2.673e−09 +/− 1.02e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #85
time = 07:06:21
pos. = 1.05 to 1.03 mm
gap = 0.799 mm
(5.12 sigma)
J = 0.00408 um
D = 1.187e−09 +/− 5.87e−10
kDAi = 3.169e−06 +/− 3.7e−07
kDAf = 5.524e−06 +/− 0
Chi2 / dof = 9.49e+04 / 373 = 254
Dg.tr = 8.165e−10 +/− 5.95e−10
Step #86
time = 07:06:46
pos. = 1.03 to 1.01 mm
gap = 0.779 mm
(4.99 sigma)
J = 0.00388 um
D = 9.656e−10 +/− 4.90e−10
kDAi = 3.967e−06 +/− 3.6e−07
kDAf = 6.319e−06 +/− 0
Chi2 / dof = 4.92e+04 / 411 = 120
Dg.tr = 1.434e−09 +/− 9.38e−10
Step #87
time = 07:07:11
pos. = 1.01 to 0.989 mm
gap = 0.759 mm
(4.87 sigma)
J = 0.00368 um
D = 1.474e−09 +/− 9.86e−10
kDAi = 4.227e−06 +/− 5.8e−07
kDAf = 6.496e−06 +/− 0
Chi2 / dof = 5.74e+04 / 404 = 142
Dg.tr = 5.283e−10 +/− 6.95e−10
Step #88
time = 07:07:36
pos. = 0.989 to 0.969 mm
gap = 0.739 mm
(4.74 sigma)
J = 0.00349 um
D = 5.378e−10 +/− 2.23e−10
kDAi = 4.284e−06 +/− 3.6e−07
kDAf = 7.291e−06 +/− 0
Chi2 / dof = 6.65e+04 / 409 = 163
Time [s]
Loss rate [Gy/s]

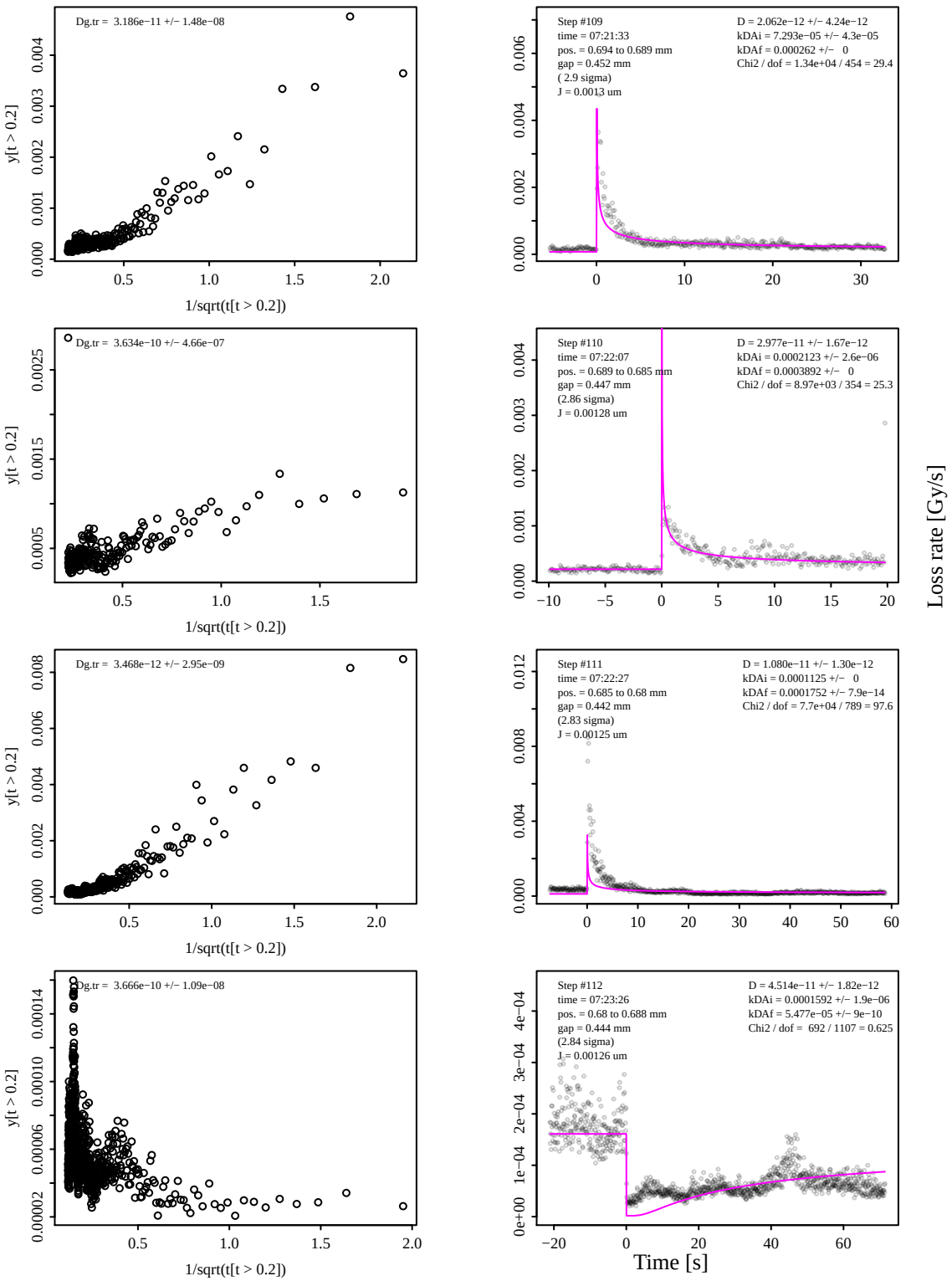

Dg.tr = 7.403e−10 +/− 7.74e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #89
time = 07:08:01
pos. = 0.969 to 0.95 mm
gap = 0.719 mm
(4.61 sigma)
J = 0.00331 um
D = 2.285e−10 +/− 4.98e−10
kDAi = 3.591e−06 +/− 2.1e−06
kDAf = 8.727e−06 +/− 0
Chi2 / dof = 2.4e+05 / 392 = 612
Dg.tr = 5.386e−10 +/− 1.55e−09
Step #90
time = 07:08:26
pos. = 0.95 to 0.93 mm
gap = 0.7 mm
(4.48 sigma)
J = 0.00313 um
D = 4.233e−10 +/− 4.79e−11
kDAi = 6.64e−06 +/− 1.6e−07
kDAf = 1.308e−05 +/− 0
Chi2 / dof = 7.67e+04 / 321 = 239
Dg.tr = 3.213e−10 +/− 3.97e−10
Step #91
time = 07:08:51
pos. = 0.93 to 0.91 mm
gap = 0.68 mm
(4.36 sigma)
J = 0.00295 um
D = 1.653e−10 +/− 1.09e−11
kDAi = 4.206e−06 +/− 0
kDAf = 6.92e−06 +/− 3.9e−06
Chi2 / dof = 1.5e+04 / 648 = 23.1
Dg.tr = 5.123e−10 +/− 1.50e−09
Step #92
time = 07:09:35
pos. = 0.91 to 0.89 mm
gap = 0.66 mm
(4.23 sigma)
J = 0.00278 um
D = 2.650e−10 +/− 3.85e−11
kDAi = 7.095e−06 +/− 1.6e−07
kDAf = 1.622e−05 +/− 0
Chi2 / dof = 6.01e+05 / 495 = 1.21e+03
Loss rate [Gy/s]
Time [s]

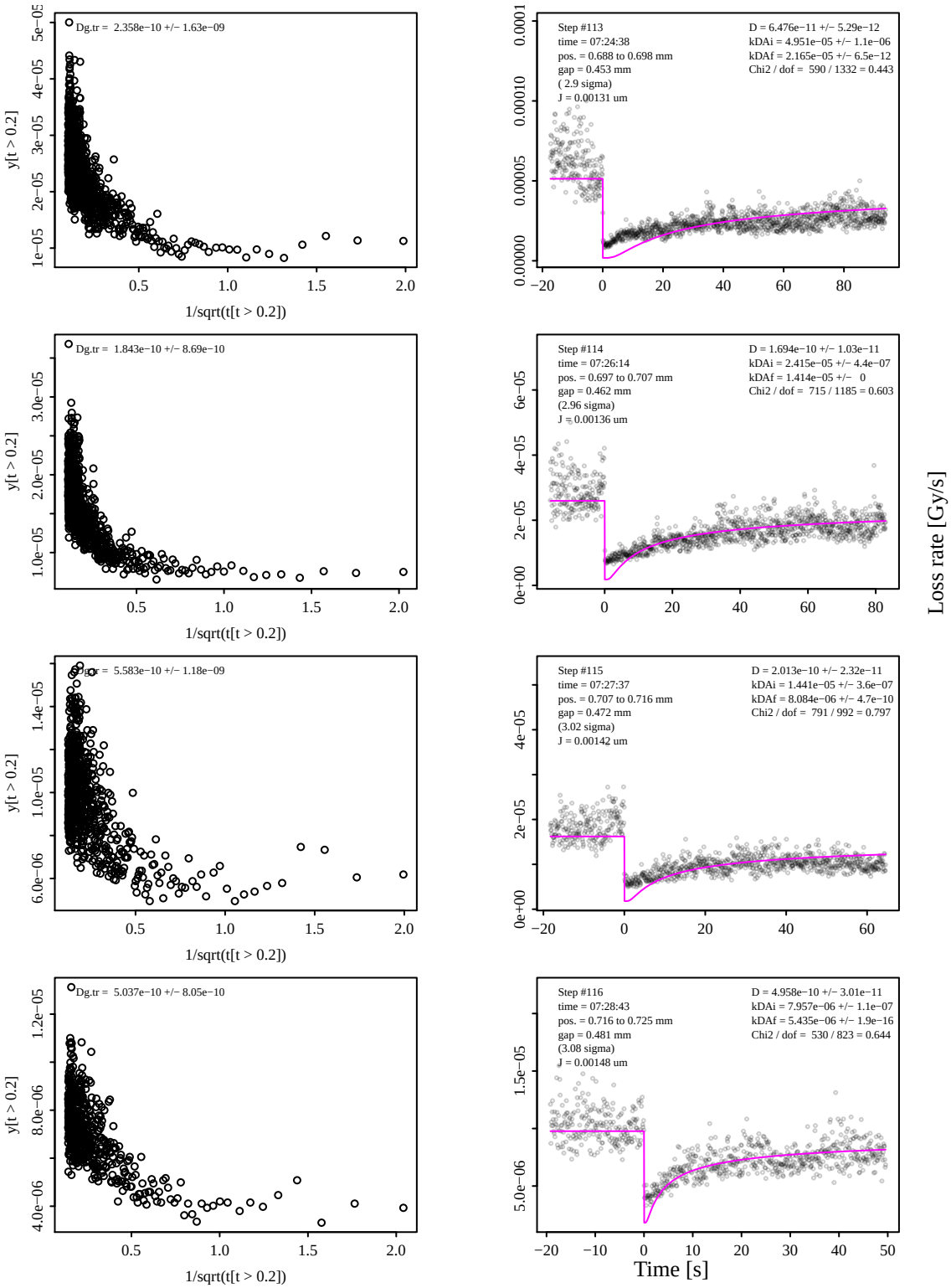

Dg.tr = 4.284e−10 +/− 1.09e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #93
time = 07:10:00
pos. = 0.89 to 0.87 mm
gap = 0.64 mm
(4.1 sigma)
J = 0.00262 um
D = 2.576e−11 +/− 9.39e−12
kDAi = 3.785e−06 +/− 4.1e−07
kDAf = 1.692e−05 +/− 0
Chi2 / dof = 3.69e+05 / 659 = 560
Dg.tr = 3.274e−10 +/− 1.00e−09
Step #94
time = 07:10:47
pos. = 0.87 to 0.85 mm
gap = 0.62 mm
(3.97 sigma)
J = 0.00246 um
D = 2.218e−10 +/− 4.08e−11
kDAi = 9.02e−06 +/− 3.8e−07
kDAf = 1.887e−05 +/− 0
Chi2 / dof = 2.52e+05 / 648 = 388
Dg.tr = 1.759e−10 +/− 1.14e−09
Step #95
time = 07:11:27
pos. = 0.85 to 0.831 mm
gap = 0.6 mm
(3.85 sigma)
J = 0.0023 um
D = 6.906e−11 +/− 2.74e−11
kDAi = 1.06e−05 +/− 1e−06
kDAf = 2.773e−05 +/− 0
Chi2 / dof = 1.69e+06 / 635 = 2.66e+03
Dg.tr = 1.425e−10 +/− 9.15e−10
Step #96
time = 07:12:07
pos. = 0.831 to 0.81 mm
gap = 0.58 mm
(3.72 sigma)
J = 0.00215 um
D = 7.177e−12 +/− 1.70e−13
kDAi = 4.947e−06 +/− 0
kDAf = 3.08e−05 +/− 0
Chi2 / dof = 1.71e+06 / 886 = 1.93e+03
Loss rate [Gy/s]
Time [s]

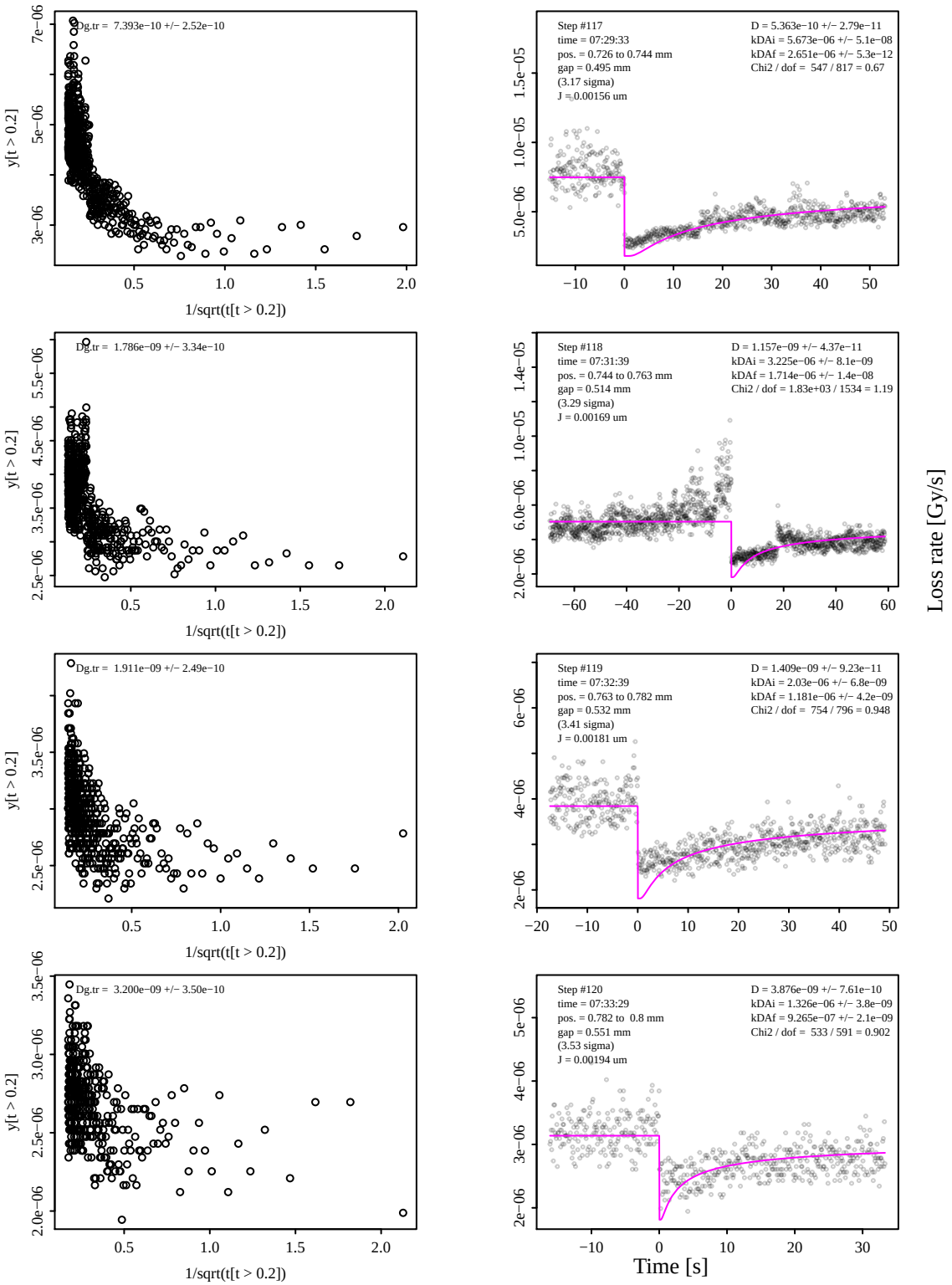

Dg.tr = 2.426e−10 +/− 4.95e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #97
time = 07:13:11
pos. = 0.81 to 0.801 mm
gap = 0.565 mm
(3.62 sigma)
J = 0.00204 um
D = 4.855e−11 +/− 1.62e−11
kDAi = 1.755e−05 +/− 1.3e−06
kDAf = 3.641e−05 +/− 0
Chi2 / dof = 1.05e+04 / 659 = 15.9
Dg.tr = 1.725e−11 +/− 1.06e−09
Step #98
time = 07:13:51
pos. = 0.801 to 0.791 mm
gap = 0.556 mm
(3.56 sigma)
J = 0.00197 um
D = 4.682e−11 +/− 1.64e−11
kDAi = 2.485e−05 +/− 1.6e−06
kDAf = 3.898e−05 +/− 4.1e−11
Chi2 / dof = 1.54e+04 / 700 = 22
Dg.tr = 4.155e−11 +/− 1.57e−09
Step #99
time = 07:14:31
pos. = 0.791 to 0.781 mm
gap = 0.546 mm
(3.5 sigma)
J = 0.0019 um
D = 2.693e−11 +/− 2.32e−11
kDAi = 1.9e−05 +/− 4.3e−06
kDAf = 4.046e−05 +/− 0
Chi2 / dof = 1.05e+05 / 629 = 167
Dg.tr = 2.667e−10 +/− 6.74e−09
Step #100
time = 07:15:11
pos. = 0.781 to 0.76 mm
gap = 0.53 mm
(3.4 sigma)
J = 0.0018 um
D = 1.771e−11 +/− 7.87e−12
kDAi = 2.148e−05 +/− 3.1e−06
kDAf = 0.0001027 +/− 0
Chi2 / dof = 1.25e+07 / 577 = 2.17e+04
Loss rate [Gy/s]
Time [s]

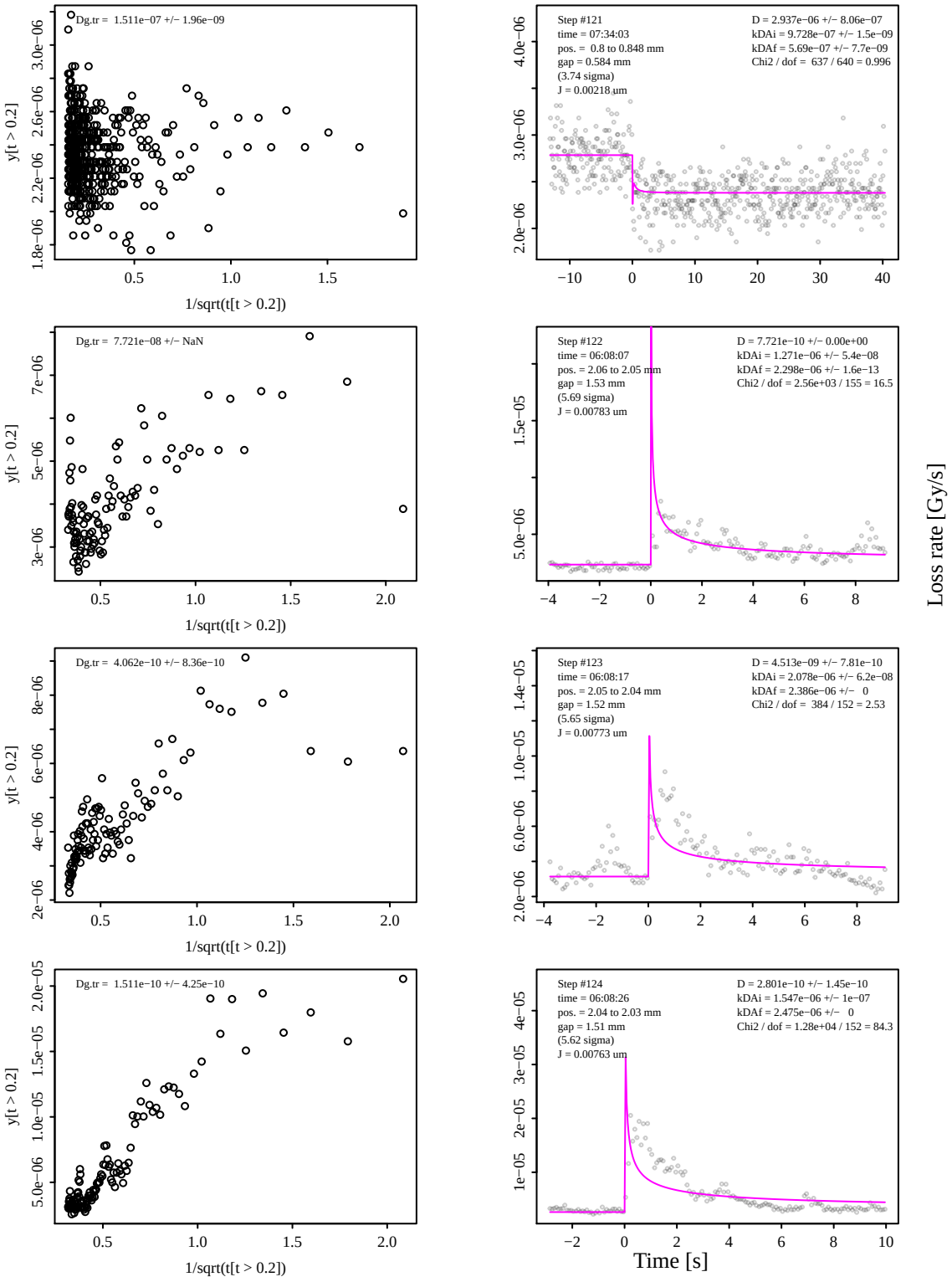

Dg.tr = 5.822e−11 +/− 2.87e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #101
time = 07:15:51
pos. = 0.761 to 0.747 mm
gap = 0.514 mm
(3.29 sigma)
J = 0.00169 um
D = 7.886e−12 +/− 6.47e−13
kDAi = 2.297e−05 +/− 0
kDAf = 9.722e−05 +/− 0
Chi2 / dof = 1.37e+05 / 582 = 235
Dg.tr = 6.962e−11 +/− 8.86e−09
Step #102
time = 07:16:31
pos. = 0.747 to 0.736 mm
gap = 0.502 mm
(3.21 sigma)
J = 0.00161 um
D = 4.206e−11 +/− 1.23e−11
kDAi = 6.022e−05 +/− 3.4e−06
kDAf = 0.0001211 +/− 0
Chi2 / dof = 1.24e+05 / 638 = 195
Dg.tr = 1.346e−11 +/− 3.00e−09
Step #103
time = 07:17:11
pos. = 0.736 to 0.726 mm
gap = 0.491 mm
(3.15 sigma)
J = 0.00154 um
D = 2.317e−12 +/− 2.53e−13
kDAi = 3.024e−05 +/− 0
kDAf = 0.0001388 +/− 0
Chi2 / dof = 3.44e+04 / 585 = 58.8
Dg.tr = 3.002e−11 +/− 2.62e−09
Step #104
time = 07:17:51
pos. = 0.726 to 0.716 mm
gap = 0.481 mm
(3.08 sigma)
J = 0.00148 um
D = 2.402e−11 +/− 2.82e−11
kDAi = 5.164e−05 +/− 1.3e−05
kDAf = 0.0001273 +/− 8.6e−05
Chi2 / dof = 1.24e+05 / 1479 = 83.7
Loss rate [Gy/s]
Time [s]

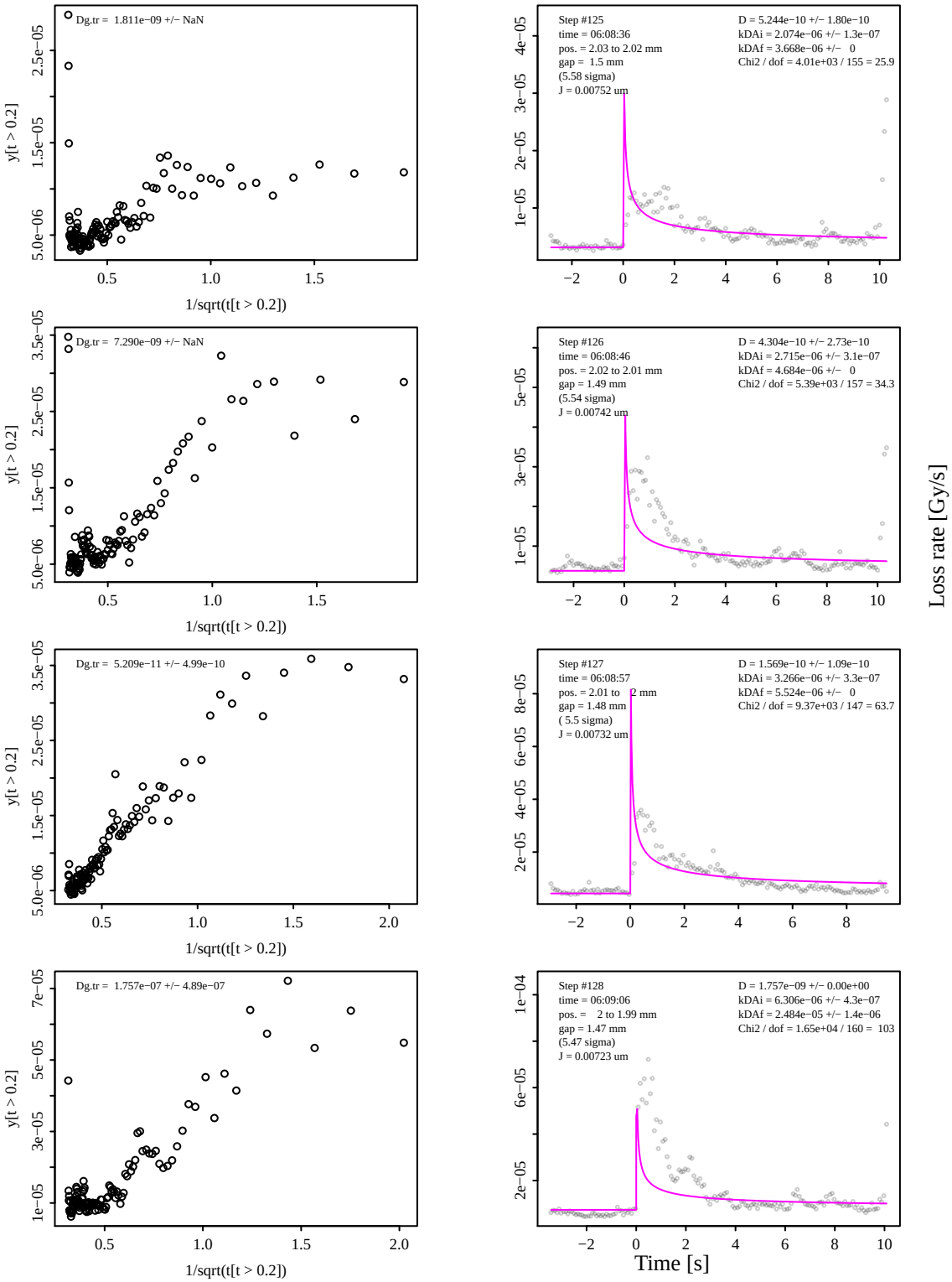

Dg.tr = 2.266e−11 +/− 9.30e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #105
time = 07:19:43
pos. = 0.716 to 0.71 mm
gap = 0.473 mm
(3.03 sigma)
J = 0.00143 um
D = 8.078e−12 +/− 1.39e−12
kDAi = 7.698e−05 +/− 2.7e−06
kDAf = 0.0001814 +/− 0
Chi2 / dof = 8.13e+03 / 666 = 12.2
Dg.tr = 4.978e−11 +/− 1.58e−08
Step #106
time = 07:20:13
pos. = 0.71 to 0.706 mm
gap = 0.468 mm
(3 sigma)
J = 0.0014 um
D = 3.368e−11 +/− 5.09e−12
kDAi = 0.0001117 +/− 3.9e−06
kDAf = 0.0001757 +/− 0
Chi2 / dof = 1.64e+03 / 455 = 3.61
Dg.tr = 6.786e−12 +/− 4.04e−09
Step #107
time = 07:20:41
pos. = 0.706 to 0.7 mm
gap = 0.463 mm
(2.97 sigma)
J = 0.00137 um
D = 2.181e−12 +/− 2.01e−13
kDAi = 5.061e−05 +/− 0
kDAf = 0.0001691 +/− 2.2e−13
Chi2 / dof = 2.2e+04 / 518 = 42.4
Dg.tr = 2.447e−11 +/− 1.59e−08
Step #108
time = 07:21:15
pos. = 0.7 to 0.695 mm
gap = 0.457 mm
(2.93 sigma)
J = 0.00134 um
D = 2.087e−11 +/− 4.25e−12
kDAi = 9.879e−05 +/− 2.6e−06
kDAf = 0.0002018 +/− 0
Chi2 / dof = 8.1e+03 / 329 = 24.6
Loss rate [Gy/s]
Time [s]

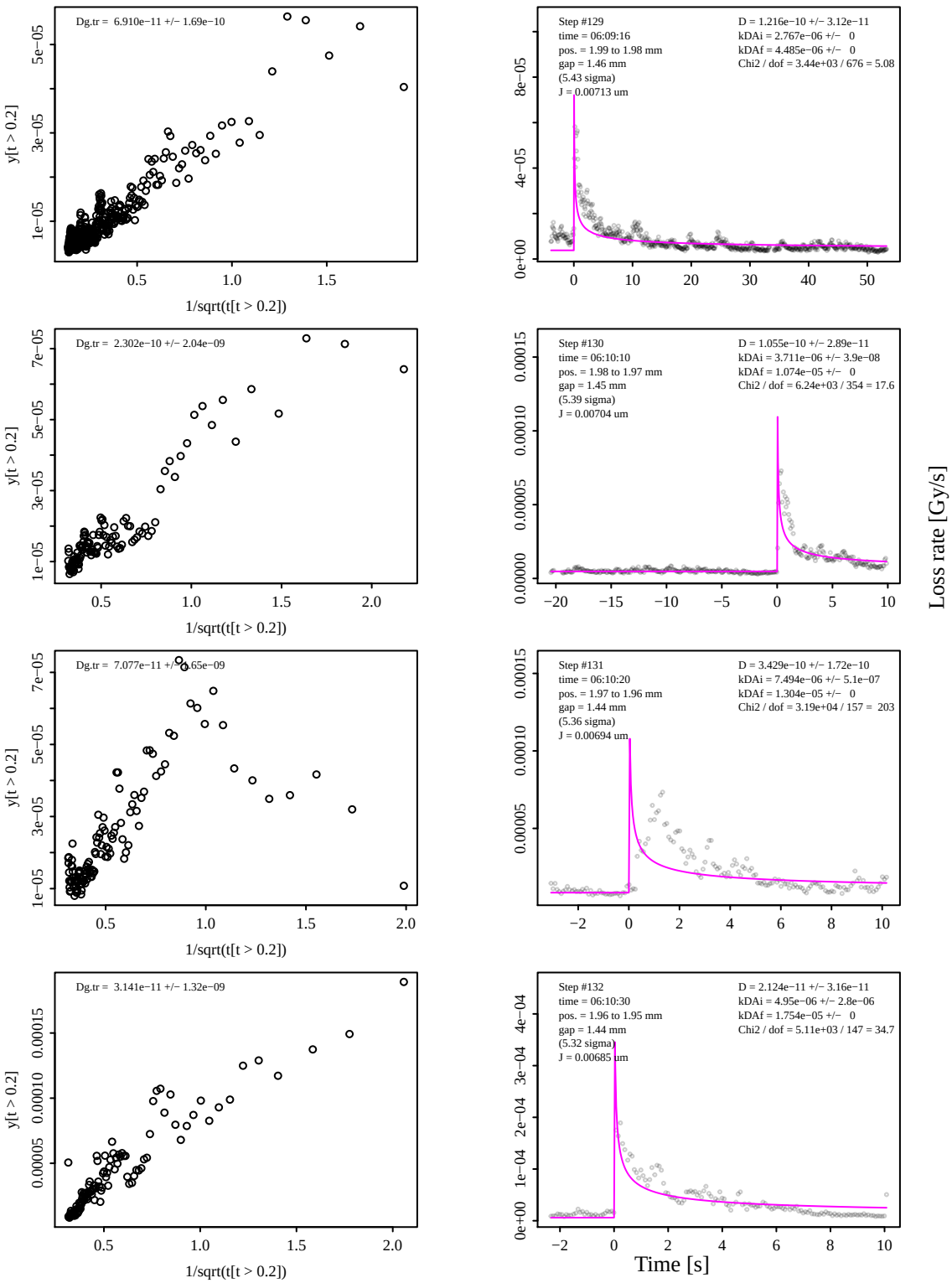

Dg.tr = 3.186e−11 +/− 1.48e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #109
time = 07:21:33
pos. = 0.694 to 0.689 mm
gap = 0.452 mm
(2.9 sigma)
J = 0.0013 um
D = 2.062e−12 +/− 4.24e−12
kDAi = 7.293e−05 +/− 4.3e−05
kDAf = 0.000262 +/− 0
Chi2 / dof = 1.34e+04 / 454 = 29.4
Dg.tr = 3.634e−10 +/− 4.66e−07
Step #110
time = 07:22:07
pos. = 0.689 to 0.685 mm
gap = 0.447 mm
(2.86 sigma)
J = 0.00128 um
D = 2.977e−11 +/− 1.67e−12
kDAi = 0.0002123 +/− 2.6e−06
kDAf = 0.0003892 +/− 0
Chi2 / dof = 8.97e+03 / 354 = 25.3
Loss rate [Gy/s]
Dg.tr = 3.468e−12 +/− 2.95e−09
Step #111
time = 07:22:27
pos. = 0.685 to 0.68 mm
gap = 0.442 mm
(2.83 sigma)
J = 0.00125 um
D = 1.080e−11 +/− 1.30e−12
kDAi = 0.0001125 +/− 0
kDAf = 0.0001752 +/− 7.9e−14
Chi2 / dof = 7.7e+04 / 789 = 97.6
Dg.tr = 3.666e−10 +/− 1.09e−08
Step #112
time = 07:23:26
pos. = 0.68 to 0.688 mm
gap = 0.444 mm
(2.84 sigma)
J = 0.00126 um
D = 4.514e−11 +/− 1.82e−12
kDAi = 0.0001592 +/− 1.9e−06
kDAf = 5.477e−05 +/− 9e−10
Chi2 / dof = 692 / 1107 = 0.625
Time [s]

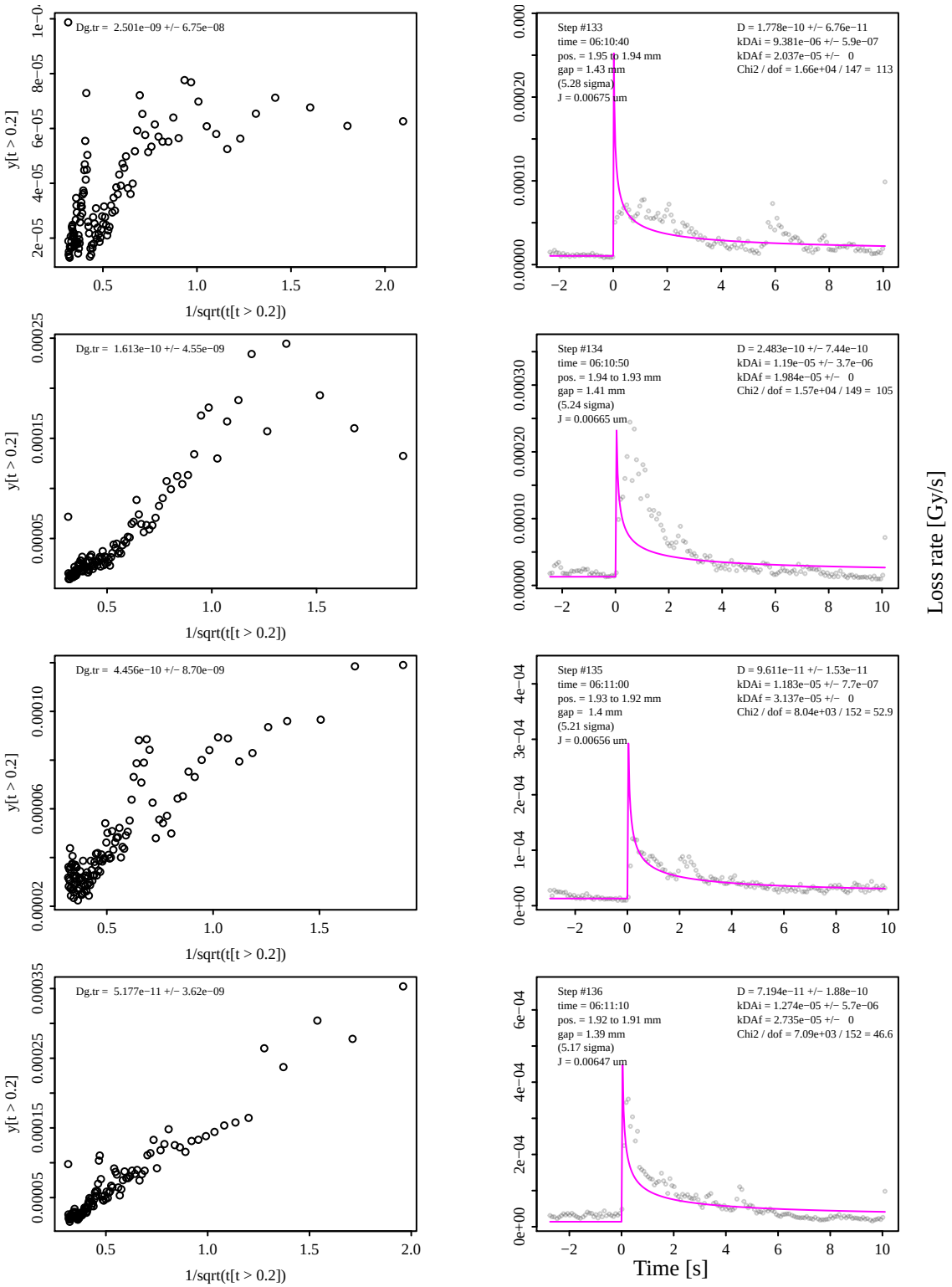

Dg.tr = 2.358e−10 +/− 1.63e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #113
time = 07:24:38
pos. = 0.688 to 0.698 mm
gap = 0.453 mm
(2.9 sigma)
J = 0.00131 um
D = 6.476e−11 +/− 5.29e−12
kDAi = 4.951e−05 +/− 1.1e−06
kDAf = 2.165e−05 +/− 6.5e−12
Chi2 / dof = 590 / 1332 = 0.443
Dg.tr = 1.843e−10 +/− 8.69e−10
Step #114
time = 07:26:14
pos. = 0.697 to 0.707 mm
gap = 0.462 mm
(2.96 sigma)
J = 0.00136 um
D = 1.694e−10 +/− 1.03e−11
kDAi = 2.415e−05 +/− 4.4e−07
kDAf = 1.414e−05 +/− 0
Chi2 / dof = 715 / 1185 = 0.603
Dg.tr = 5.583e−10 +/− 1.18e−09
Step #115
time = 07:27:37
pos. = 0.707 to 0.716 mm
gap = 0.472 mm
(3.02 sigma)
J = 0.00142 um
D = 2.013e−10 +/− 2.32e−11
kDAi = 1.441e−05 +/− 3.6e−07
kDAf = 8.084e−06 +/− 4.7e−10
Chi2 / dof = 791 / 992 = 0.797
Dg.tr = 5.037e−10 +/− 8.05e−10
Step #116
time = 07:28:43
pos. = 0.716 to 0.725 mm
gap = 0.481 mm
(3.08 sigma)
J = 0.00148 um
D = 4.958e−10 +/− 3.01e−11
kDAi = 7.957e−06 +/− 1.1e−07
kDAf = 5.435e−06 +/− 1.9e−16
Chi2 / dof = 530 / 823 = 0.644
Loss rate [Gy/s]
Time [s]

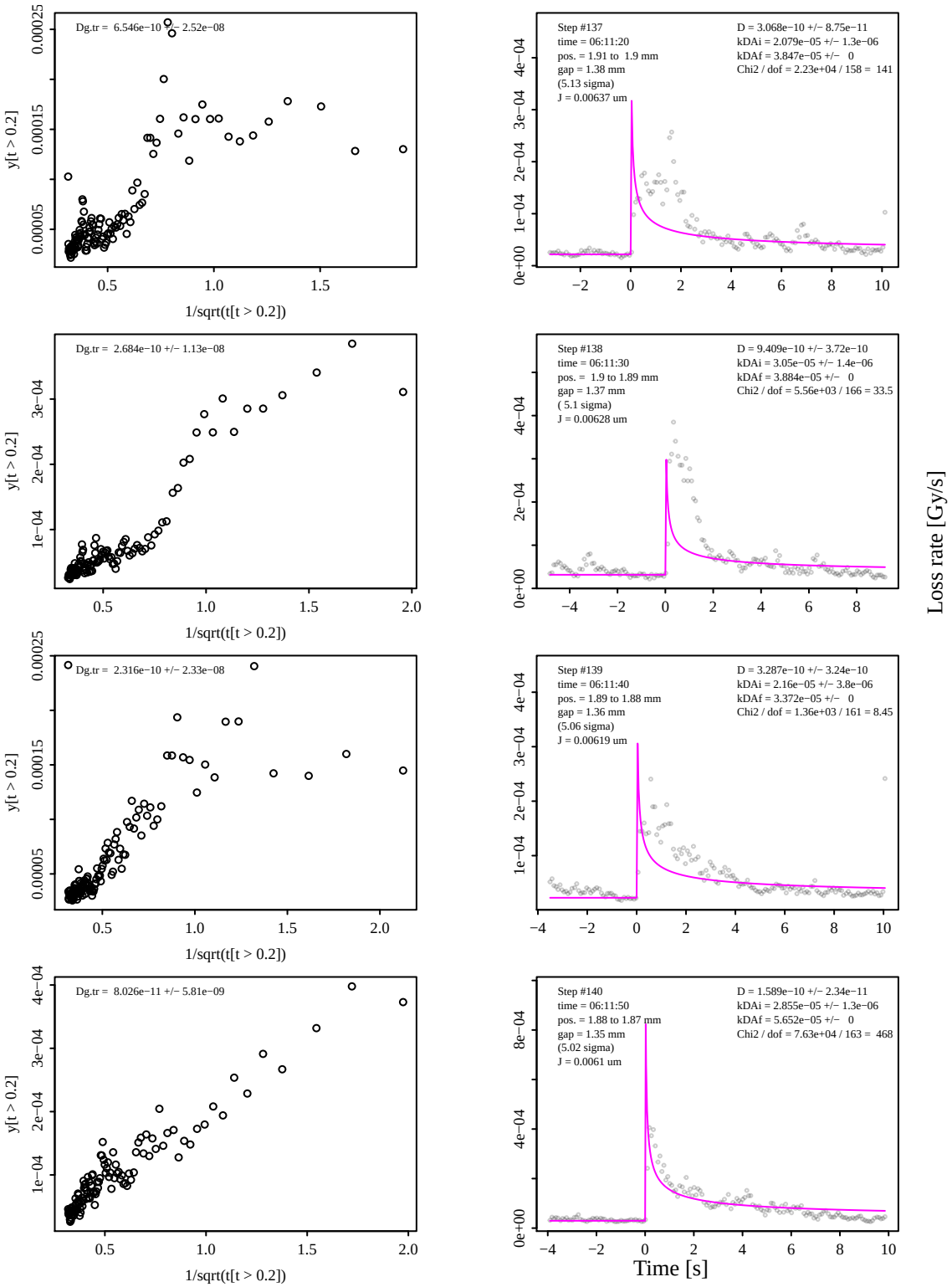

Dg.tr = 7.393e−10 +/− 2.52e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #117
time = 07:29:33
pos. = 0.726 to 0.744 mm
gap = 0.495 mm
(3.17 sigma)
J = 0.00156 um
D = 5.363e−10 +/− 2.79e−11
kDAi = 5.673e−06 +/− 5.1e−08
kDAf = 2.651e−06 +/− 5.3e−12
Chi2 / dof = 547 / 817 = 0.67
Dg.tr = 1.786e−09 +/− 3.34e−10
Step #118
time = 07:31:39
pos. = 0.744 to 0.763 mm
gap = 0.514 mm
(3.29 sigma)
J = 0.00169 um
D = 1.157e−09 +/− 4.37e−11
kDAi = 3.225e−06 +/− 8.1e−09
kDAf = 1.714e−06 +/− 1.4e−08
Chi2 / dof = 1.83e+03 / 1534 = 1.19
Dg.tr = 1.911e−09 +/− 2.49e−10
Step #119
time = 07:32:39
pos. = 0.763 to 0.782 mm
gap = 0.532 mm
(3.41 sigma)
J = 0.00181 um
D = 1.409e−09 +/− 9.23e−11
kDAi = 2.03e−06 +/− 6.8e−09
kDAf = 1.181e−06 +/− 4.2e−09
Chi2 / dof = 754 / 796 = 0.948
Dg.tr = 3.200e−09 +/− 3.50e−10
Step #120
time = 07:33:29
pos. = 0.782 to 0.8 mm
gap = 0.551 mm
(3.53 sigma)
J = 0.00194 um
D = 3.876e−09 +/− 7.61e−10
kDAi = 1.326e−06 +/− 3.8e−09
kDAf = 9.265e−07 +/− 2.1e−09
Chi2 / dof = 533 / 591 = 0.902
Loss rate [Gy/s]
Time [s]

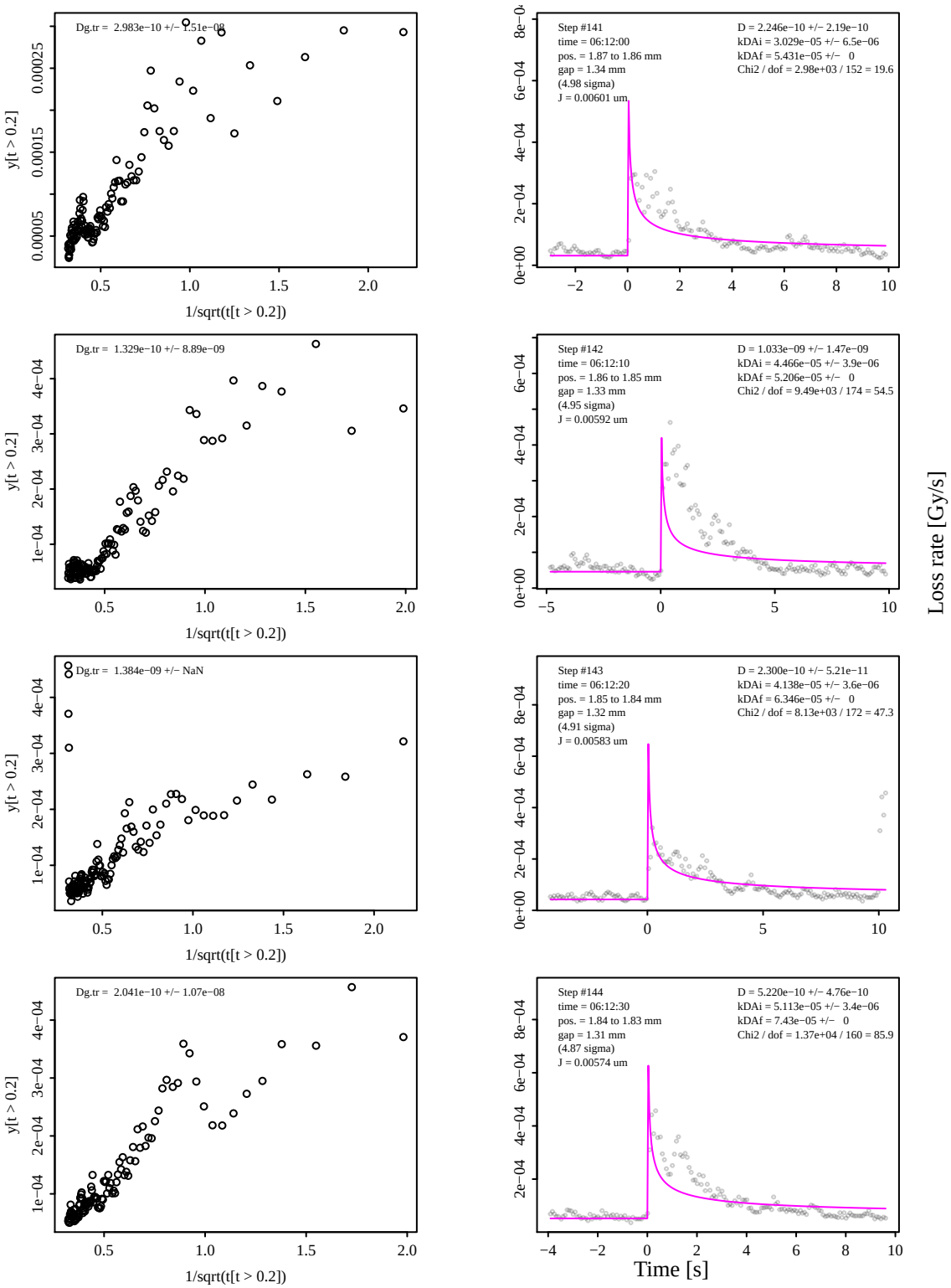

Dg.tr = 1.511e−07 +/− 1.96e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #121
time = 07:34:03
pos. = 0.8 to 0.848 mm
gap = 0.584 mm
(3.74 sigma)
J = 0.00218 um
D = 2.937e−06 +/− 8.06e−07
kDAi = 9.728e−07 +/− 1.5e−09
kDAf = 5.69e−07 +/− 7.7e−09
Chi2 / dof = 637 / 640 = 0.996
Dg.tr = 7.721e−08 +/− NaN
Step #122
time = 06:08:07
pos. = 2.06 to 2.05 mm
gap = 1.53 mm
(5.69 sigma)
J = 0.00783 um
D = 7.721e−10 +/− 0.00e+00
kDAi = 1.271e−06 +/− 5.4e−08
kDAf = 2.298e−06 +/− 1.6e−13
Chi2 / dof = 2.56e+03 / 155 = 16.5
Dg.tr = 4.062e−10 +/− 8.36e−10
Step #123
time = 06:08:17
pos. = 2.05 to 2.04 mm
gap = 1.52 mm
(5.65 sigma)
J = 0.00773 um
D = 4.513e−09 +/− 7.81e−10
kDAi = 2.078e−06 +/− 6.2e−08
kDAf = 2.386e−06 +/− 0
Chi2 / dof = 384 / 152 = 2.53
Dg.tr = 1.511e−10 +/− 4.25e−10
Step #124
time = 06:08:26
pos. = 2.04 to 2.03 mm
gap = 1.51 mm
(5.62 sigma)
J = 0.00763 um
D = 2.801e−10 +/− 1.45e−10
kDAi = 1.547e−06 +/− 1e−07
kDAf = 2.475e−06 +/− 0
Chi2 / dof = 1.28e+04 / 152 = 84.3
Loss rate [Gy/s]
Time [s]

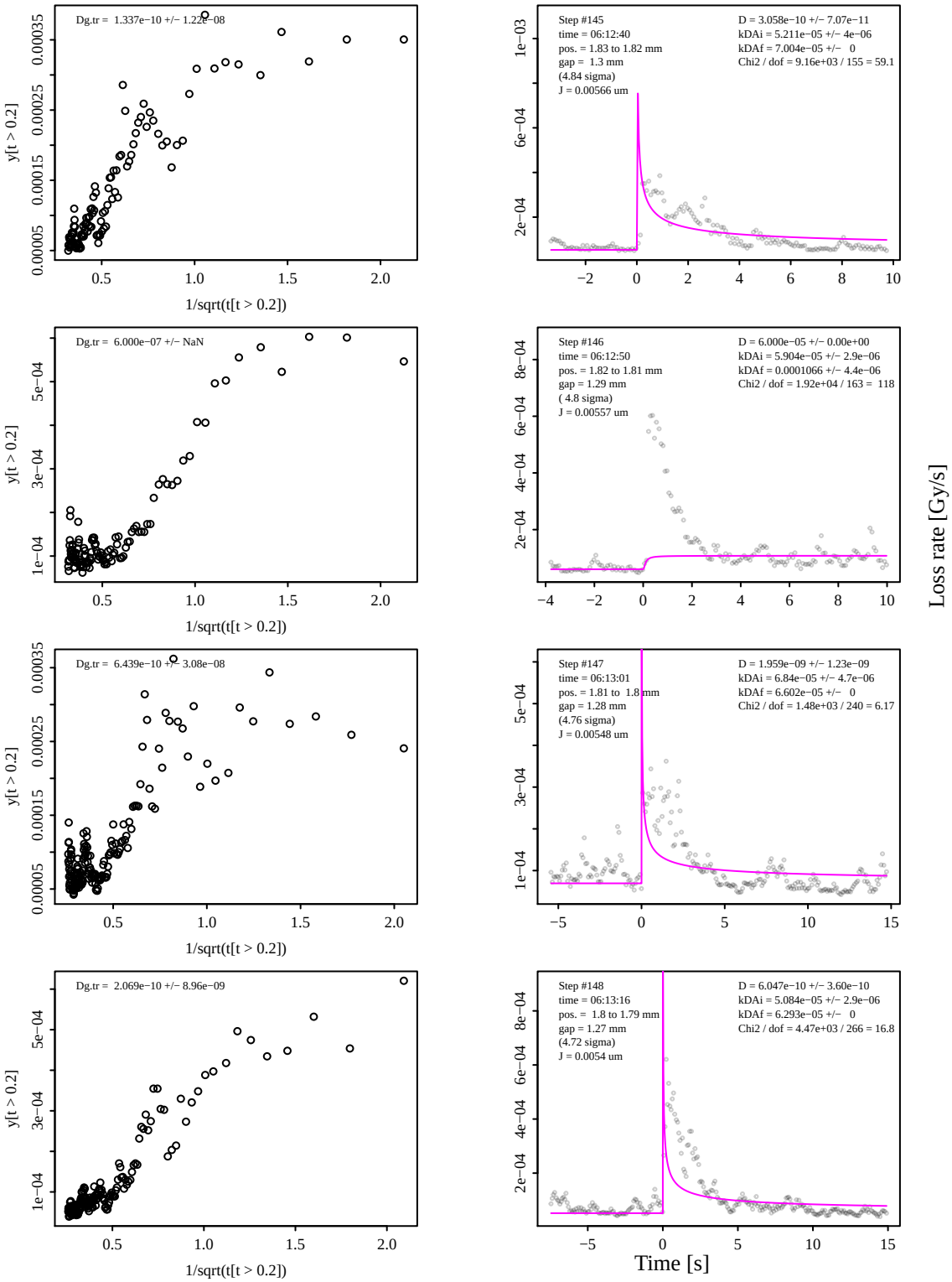

Dg.tr = 1.811e−09 +/− NaN
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #125
time = 06:08:36
pos. = 2.03 to 2.02 mm
gap = 1.5 mm
(5.58 sigma)
J = 0.00752 um
D = 5.244e−10 +/− 1.80e−10
kDAi = 2.074e−06 +/− 1.3e−07
kDAf = 3.668e−06 +/− 0
Chi2 / dof = 4.01e+03 / 155 = 25.9
Dg.tr = 7.290e−09 +/− NaN
Step #126
time = 06:08:46
pos. = 2.02 to 2.01 mm
gap = 1.49 mm
(5.54 sigma)
J = 0.00742 um
D = 4.304e−10 +/− 2.73e−10
kDAi = 2.715e−06 +/− 3.1e−07
kDAf = 4.684e−06 +/− 0
Chi2 / dof = 5.39e+03 / 157 = 34.3
Dg.tr = 5.209e−11 +/− 4.99e−10
Step #127
time = 06:08:57
pos. = 2.01 to 2 mm
gap = 1.48 mm
(5.5 sigma)
J = 0.00732 um
D = 1.569e−10 +/− 1.09e−10
kDAi = 3.266e−06 +/− 3.3e−07
kDAf = 5.524e−06 +/− 0
Chi2 / dof = 9.37e+03 / 147 = 63.7
Dg.tr = 1.757e−07 +/− 4.89e−07
Step #128
time = 06:09:06
pos. = 2 to 1.99 mm
gap = 1.47 mm
(5.47 sigma)
J = 0.00723 um
D = 1.757e−09 +/− 0.00e+00
kDAi = 6.306e−06 +/− 4.3e−07
kDAf = 2.484e−05 +/− 1.4e−06
Chi2 / dof = 1.65e+04 / 160 = 103
Loss rate [Gy/s]
Time [s]

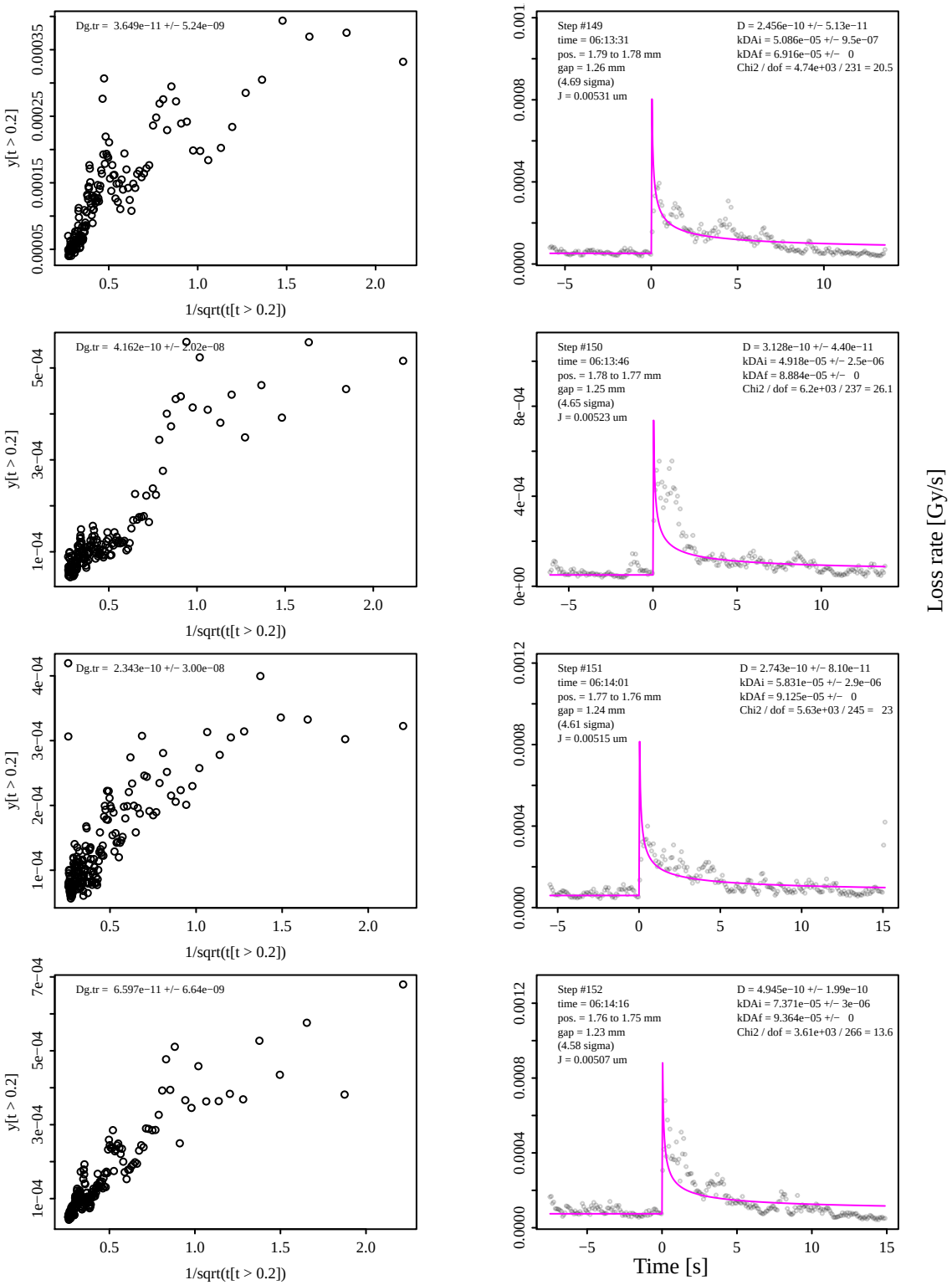

Dg.tr = 6.910e−11 +/− 1.69e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #129
time = 06:09:16
pos. = 1.99 to 1.98 mm
gap = 1.46 mm
(5.43 sigma)
J = 0.00713 um
D = 1.216e−10 +/− 3.12e−11
kDAi = 2.767e−06 +/− 0
kDAf = 4.485e−06 +/− 0
Chi2 / dof = 3.44e+03 / 676 = 5.08
Dg.tr = 2.302e−10 +/− 2.04e−09
Step #130
time = 06:10:10
pos. = 1.98 to 1.97 mm
gap = 1.45 mm
(5.39 sigma)
J = 0.00704 um
D = 1.055e−10 +/− 2.89e−11
kDAi = 3.711e−06 +/− 3.9e−08
kDAf = 1.074e−05 +/− 0
Chi2 / dof = 6.24e+03 / 354 = 17.6
Dg.tr = 7.077e−11 +/− 6.65e−09
Step #131
time = 06:10:20
pos. = 1.97 to 1.96 mm
gap = 1.44 mm
(5.36 sigma)
J = 0.00694 um
D = 3.429e−10 +/− 1.72e−10
kDAi = 7.494e−06 +/− 5.1e−07
kDAf = 1.304e−05 +/− 0
Chi2 / dof = 3.19e+04 / 157 = 203
Dg.tr = 3.141e−11 +/− 1.32e−09
Step #132
time = 06:10:30
pos. = 1.96 to 1.95 mm
gap = 1.44 mm
(5.32 sigma)
J = 0.00685 um
D = 2.124e−11 +/− 3.16e−11
kDAi = 4.95e−06 +/− 2.8e−06
kDAf = 1.754e−05 +/− 0
Chi2 / dof = 5.11e+03 / 147 = 34.7
Loss rate [Gy/s]
Time [s]

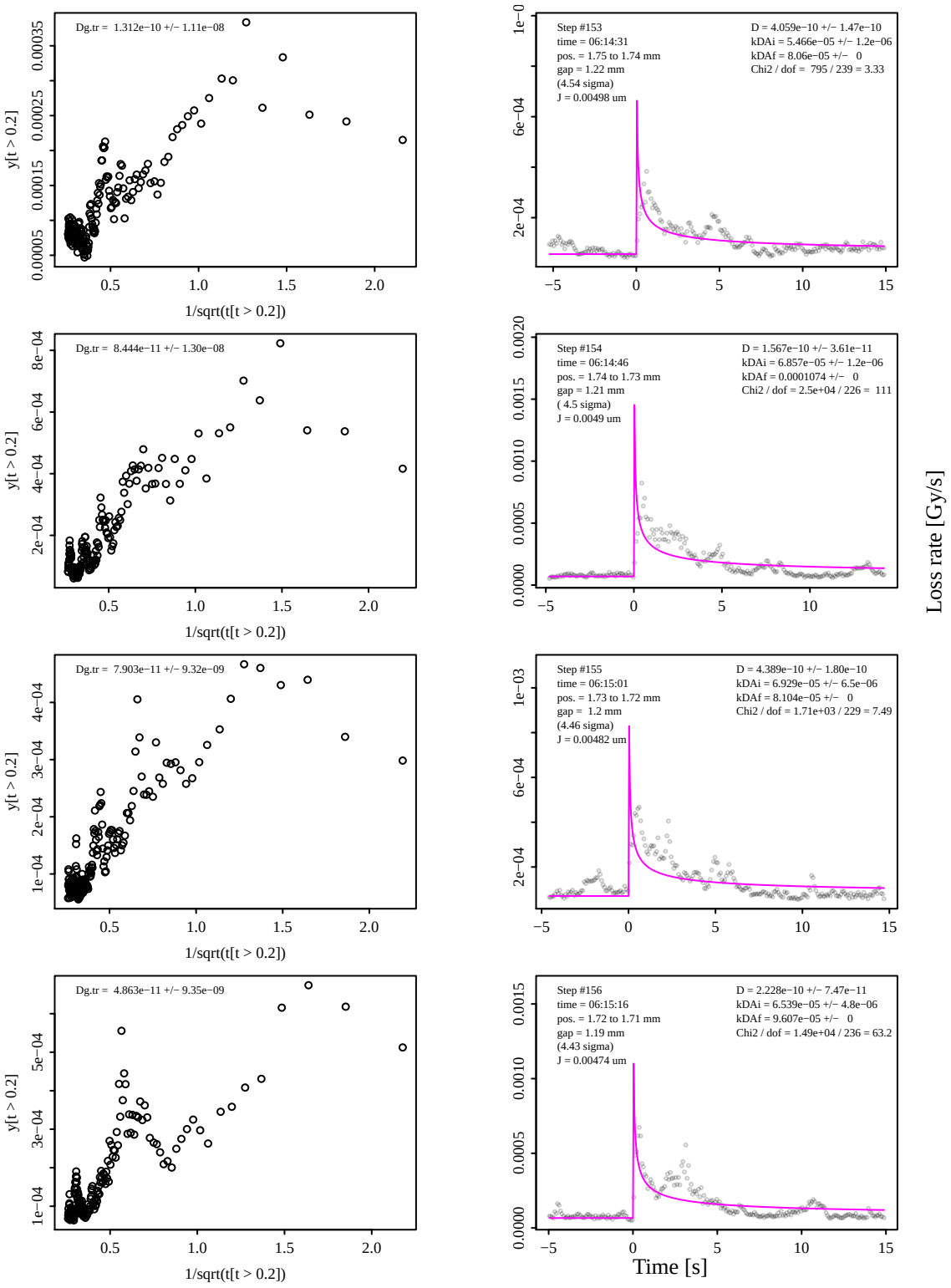

Dg.tr = 2.501e−09 +/− 6.75e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #133
time = 06:10:40
pos. = 1.95 to 1.94 mm
gap = 1.43 mm
(5.28 sigma)
J = 0.00675 um
D = 1.778e−10 +/− 6.76e−11
kDAi = 9.381e−06 +/− 5.9e−07
kDAf = 2.037e−05 +/− 0
Chi2 / dof = 1.66e+04 / 147 = 113
Dg.tr = 1.613e−10 +/− 4.55e−09
Step #134
time = 06:10:50
pos. = 1.94 to 1.93 mm
gap = 1.41 mm
(5.24 sigma)
J = 0.00665 um
D = 2.483e−10 +/− 7.44e−10
kDAi = 1.19e−05 +/− 3.7e−06
kDAf = 1.984e−05 +/− 0
Chi2 / dof = 1.57e+04 / 149 = 105
Dg.tr = 4.456e−10 +/− 8.70e−09
Step #135
time = 06:11:00
pos. = 1.93 to 1.92 mm
gap = 1.4 mm
(5.21 sigma)
J = 0.00656 um
D = 9.611e−11 +/− 1.53e−11
kDAi = 1.183e−05 +/− 7.7e−07
kDAf = 3.137e−05 +/− 0
Chi2 / dof = 8.04e+03 / 152 = 52.9
Dg.tr = 5.177e−11 +/− 3.62e−09
Step #136
time = 06:11:10
pos. = 1.92 to 1.91 mm
gap = 1.39 mm
(5.17 sigma)
J = 0.00647 um
D = 7.194e−11 +/− 1.88e−10
kDAi = 1.274e−05 +/− 5.7e−06
kDAf = 2.735e−05 +/− 0
Chi2 / dof = 7.09e+03 / 152 = 46.6
Loss rate [Gy/s]
Time [s]

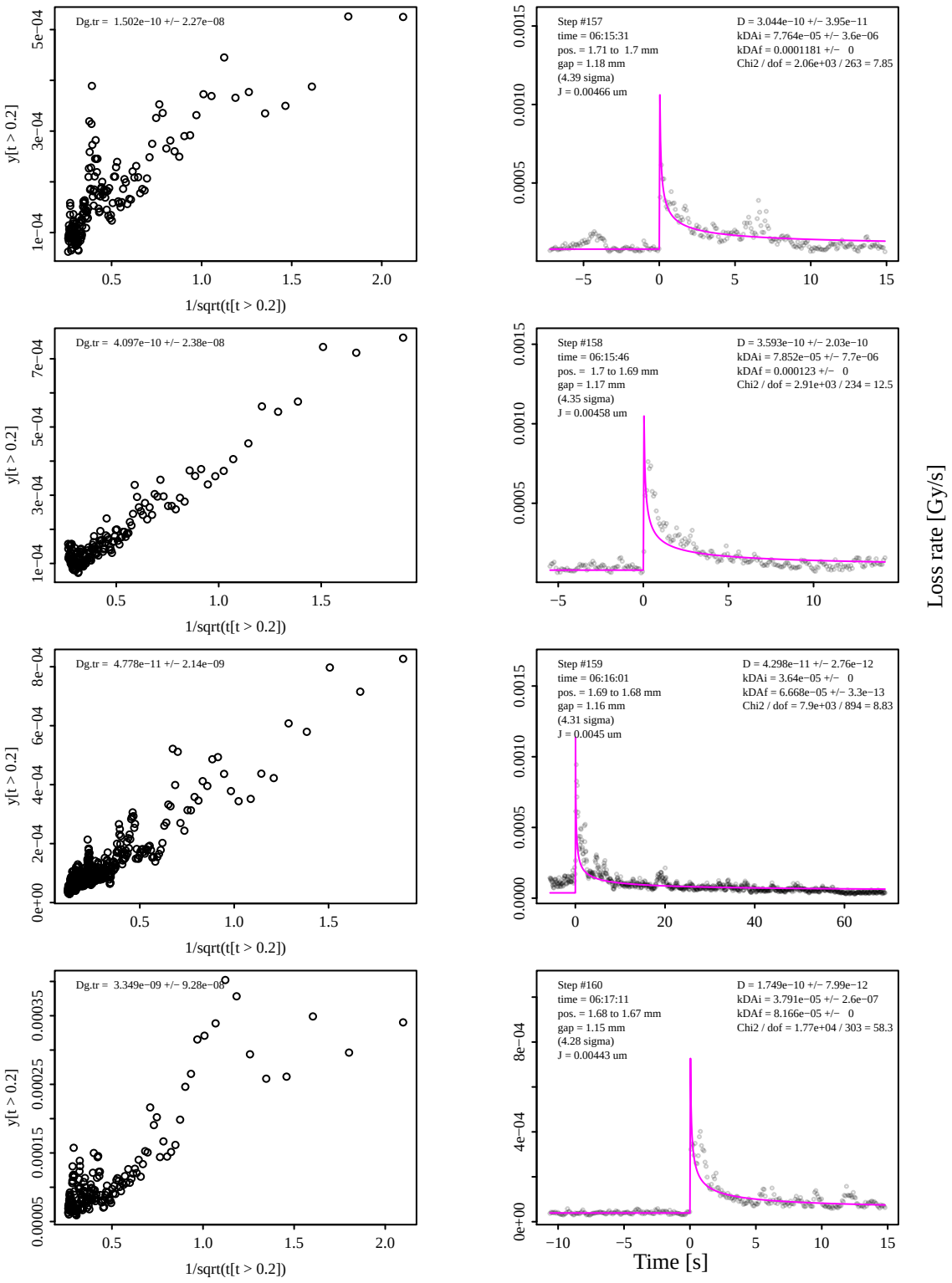

Dg.tr = 6.546e−10 +/− 2.52e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #137
time = 06:11:20
pos. = 1.91 to 1.9 mm
gap = 1.38 mm
(5.13 sigma)
J = 0.00637 um
D = 3.068e−10 +/− 8.75e−11
kDAi = 2.079e−05 +/− 1.3e−06
kDAf = 3.847e−05 +/− 0
Chi2 / dof = 2.23e+04 / 158 = 141
Dg.tr = 2.684e−10 +/− 1.13e−08
Step #138
time = 06:11:30
pos. = 1.9 to 1.89 mm
gap = 1.37 mm
(5.1 sigma)
J = 0.00628 um
D = 9.409e−10 +/− 3.72e−10
kDAi = 3.05e−05 +/− 1.4e−06
kDAf = 3.884e−05 +/− 0
Chi2 / dof = 5.56e+03 / 166 = 33.5
Dg.tr = 2.316e−10 +/− 2.33e−08
Step #139
time = 06:11:40
pos. = 1.89 to 1.88 mm
gap = 1.36 mm
(5.06 sigma)
J = 0.00619 um
D = 3.287e−10 +/− 3.24e−10
kDAi = 2.16e−05 +/− 3.8e−06
kDAf = 3.372e−05 +/− 0
Chi2 / dof = 1.36e+03 / 161 = 8.45
Dg.tr = 8.026e−11 +/− 5.81e−09
Step #140
time = 06:11:50
pos. = 1.88 to 1.87 mm
gap = 1.35 mm
(5.02 sigma)
J = 0.0061 um
D = 1.589e−10 +/− 2.34e−11
kDAi = 2.855e−05 +/− 1.3e−06
kDAf = 5.652e−05 +/− 0
Chi2 / dof = 7.63e+04 / 163 = 468
Loss rate [Gy/s]
Time [s]

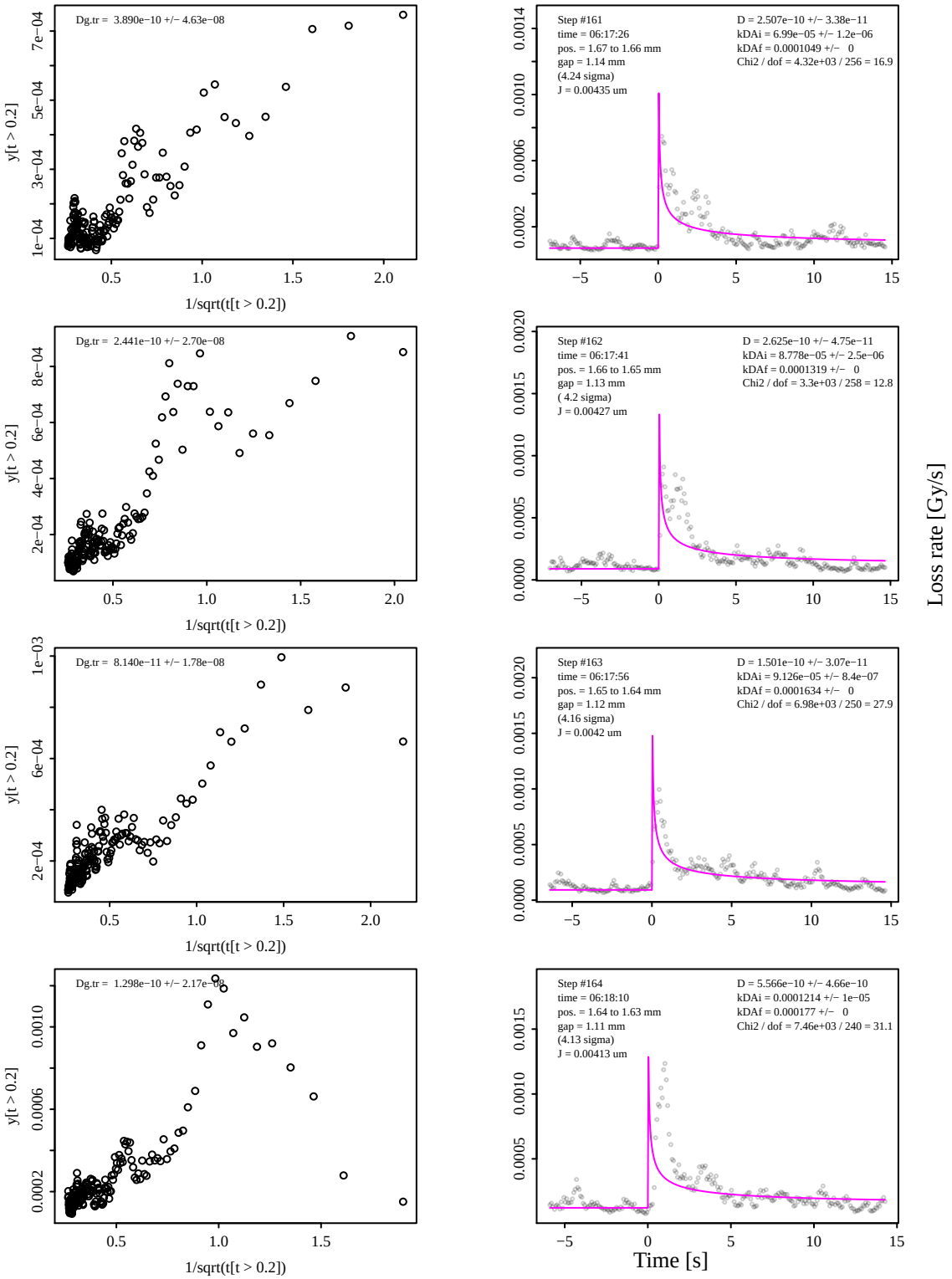

Dg.tr = 2.983e−10 +/− 1.51e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #141
time = 06:12:00
pos. = 1.87 to 1.86 mm
gap = 1.34 mm
(4.98 sigma)
J = 0.00601 um
D = 2.246e−10 +/− 2.19e−10
kDAi = 3.029e−05 +/− 6.5e−06
kDAf = 5.431e−05 +/− 0
Chi2 / dof = 2.98e+03 / 152 = 19.6
Dg.tr = 1.329e−10 +/− 8.89e−09
Step #142
time = 06:12:10
pos. = 1.86 to 1.85 mm
gap = 1.33 mm
(4.95 sigma)
J = 0.00592 um
D = 1.033e−09 +/− 1.47e−09
kDAi = 4.466e−05 +/− 3.9e−06
kDAf = 5.206e−05 +/− 0
Chi2 / dof = 9.49e+03 / 174 = 54.5
Dg.tr = 1.384e−09 +/− NaN
Step #143
time = 06:12:20
pos. = 1.85 to 1.84 mm
gap = 1.32 mm
(4.91 sigma)
J = 0.00583 um
D = 2.300e−10 +/− 5.21e−11
kDAi = 4.138e−05 +/− 3.6e−06
kDAf = 6.346e−05 +/− 0
Chi2 / dof = 8.13e+03 / 172 = 47.3
Dg.tr = 2.041e−10 +/− 1.07e−08
Step #144
time = 06:12:30
pos. = 1.84 to 1.83 mm
gap = 1.31 mm
(4.87 sigma)
J = 0.00574 um
D = 5.220e−10 +/− 4.76e−10
kDAi = 5.113e−05 +/− 3.4e−06
kDAf = 7.43e−05 +/− 0
Chi2 / dof = 1.37e+04 / 160 = 85.9
Loss rate [Gy/s]
Time [s]

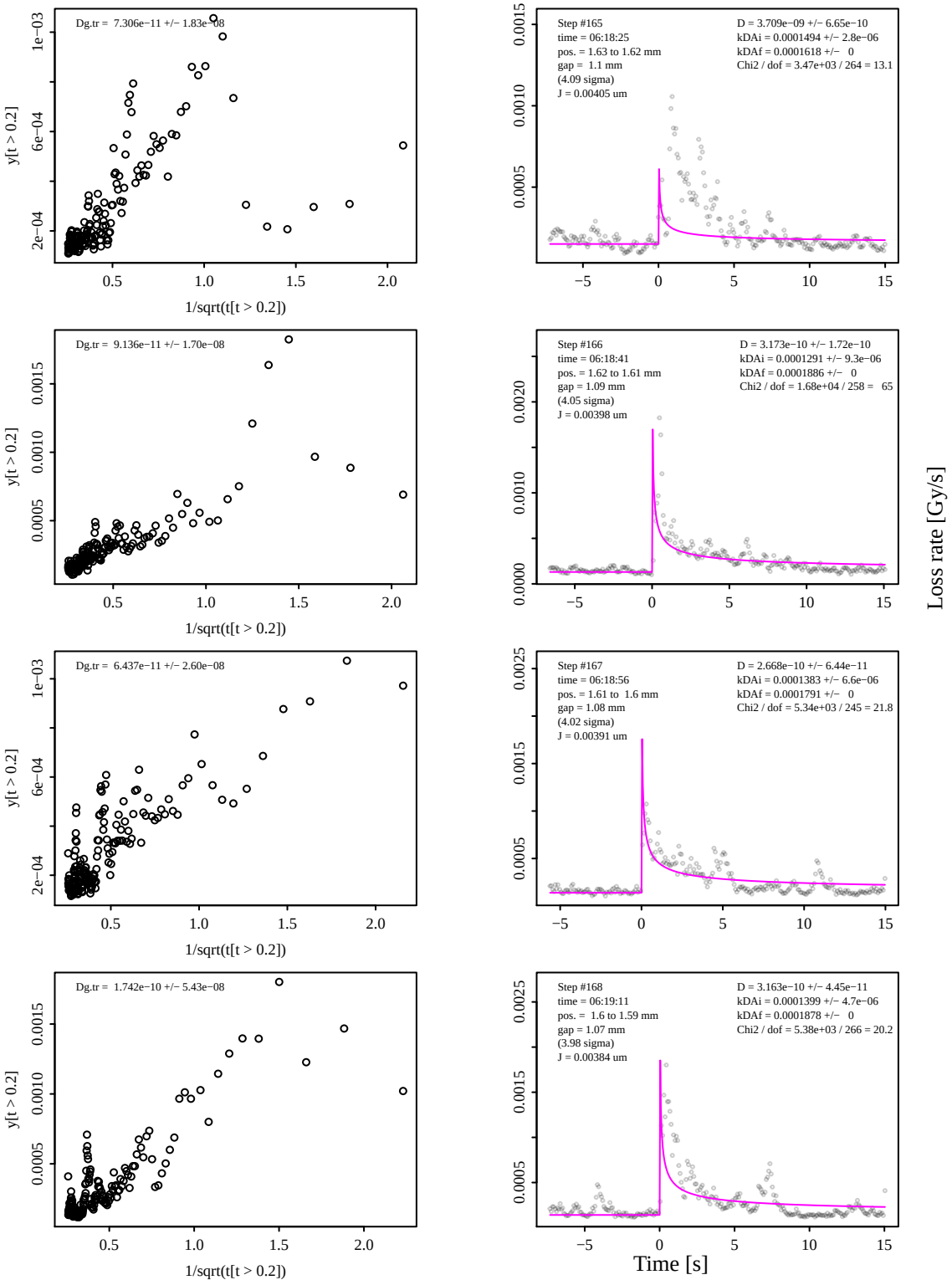

Dg.tr = 1.337e−10 +/− 1.22e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #145
time = 06:12:40
pos. = 1.83 to 1.82 mm
gap = 1.3 mm
(4.84 sigma)
J = 0.00566 um
D = 3.058e−10 +/− 7.07e−11
kDAi = 5.211e−05 +/− 4e−06
kDAf = 7.004e−05 +/− 0
Chi2 / dof = 9.16e+03 / 155 = 59.1
Dg.tr = 6.000e−07 +/− NaN
Step #146
time = 06:12:50
pos. = 1.82 to 1.81 mm
gap = 1.29 mm
(4.8 sigma)
J = 0.00557 um
D = 6.000e−05 +/− 0.00e+00
kDAi = 5.904e−05 +/− 2.9e−06
kDAf = 0.0001066 +/− 4.4e−06
Chi2 / dof = 1.92e+04 / 163 = 118
Dg.tr = 6.439e−10 +/− 3.08e−08
Step #147
time = 06:13:01
pos. = 1.81 to 1.8 mm
gap = 1.28 mm
(4.76 sigma)
J = 0.00548 um
D = 1.959e−09 +/− 1.23e−09
kDAi = 6.84e−05 +/− 4.7e−06
kDAf = 6.602e−05 +/− 0
Chi2 / dof = 1.48e+03 / 240 = 6.17
Dg.tr = 2.069e−10 +/− 8.96e−09
Step #148
time = 06:13:16
pos. = 1.8 to 1.79 mm
gap = 1.27 mm
(4.72 sigma)
J = 0.0054 um
D = 6.047e−10 +/− 3.60e−10
kDAi = 5.084e−05 +/− 2.9e−06
kDAf = 6.293e−05 +/− 0
Chi2 / dof = 4.47e+03 / 266 = 16.8
Loss rate [Gy/s]
Time [s]

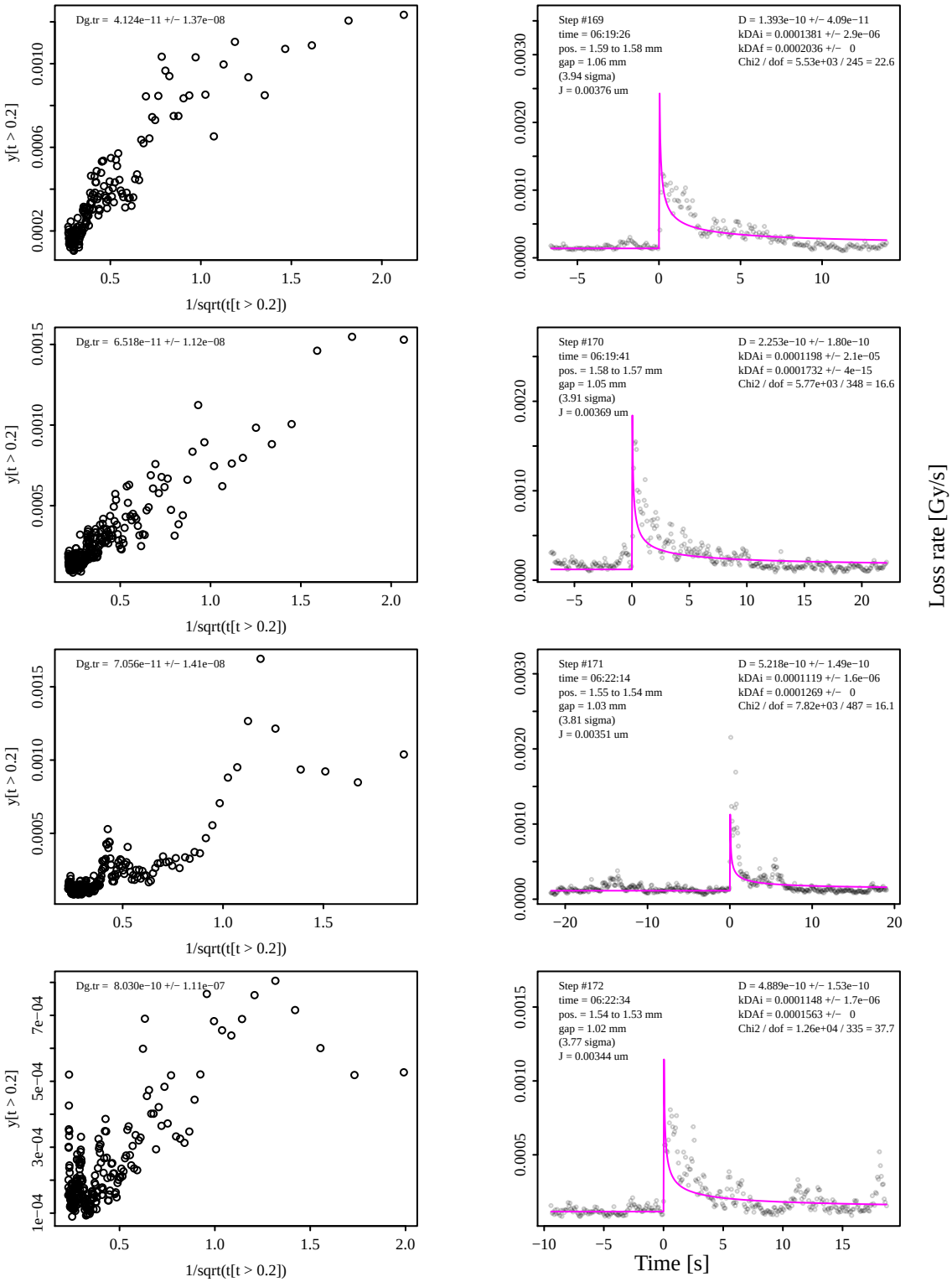

Dg.tr = 3.649e−11 +/− 5.24e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #149
time = 06:13:31
pos. = 1.79 to 1.78 mm
gap = 1.26 mm
(4.69 sigma)
J = 0.00531 um
D = 2.456e−10 +/− 5.13e−11
kDAi = 5.086e−05 +/− 9.5e−07
kDAf = 6.916e−05 +/− 0
Chi2 / dof = 4.74e+03 / 231 = 20.5
Dg.tr = 4.162e−10 +/− 2.02e−08
Step #150
time = 06:13:46
pos. = 1.78 to 1.77 mm
gap = 1.25 mm
(4.65 sigma)
J = 0.00523 um
D = 3.128e−10 +/− 4.40e−11
kDAi = 4.918e−05 +/− 2.5e−06
kDAf = 8.884e−05 +/− 0
Chi2 / dof = 6.2e+03 / 237 = 26.1
Dg.tr = 2.343e−10 +/− 3.00e−08
Step #151
time = 06:14:01
pos. = 1.77 to 1.76 mm
gap = 1.24 mm
(4.61 sigma)
J = 0.00515 um
D = 2.743e−10 +/− 8.10e−11
kDAi = 5.831e−05 +/− 2.9e−06
kDAf = 9.125e−05 +/− 0
Chi2 / dof = 5.63e+03 / 245 = 23
Dg.tr = 6.597e−11 +/− 6.64e−09
Step #152
time = 06:14:16
pos. = 1.76 to 1.75 mm
gap = 1.23 mm
(4.58 sigma)
J = 0.00507 um
D = 4.945e−10 +/− 1.99e−10
kDAi = 7.371e−05 +/− 3e−06
kDAf = 9.364e−05 +/− 0
Chi2 / dof = 3.61e+03 / 266 = 13.6
Loss rate [Gy/s]
Time [s]

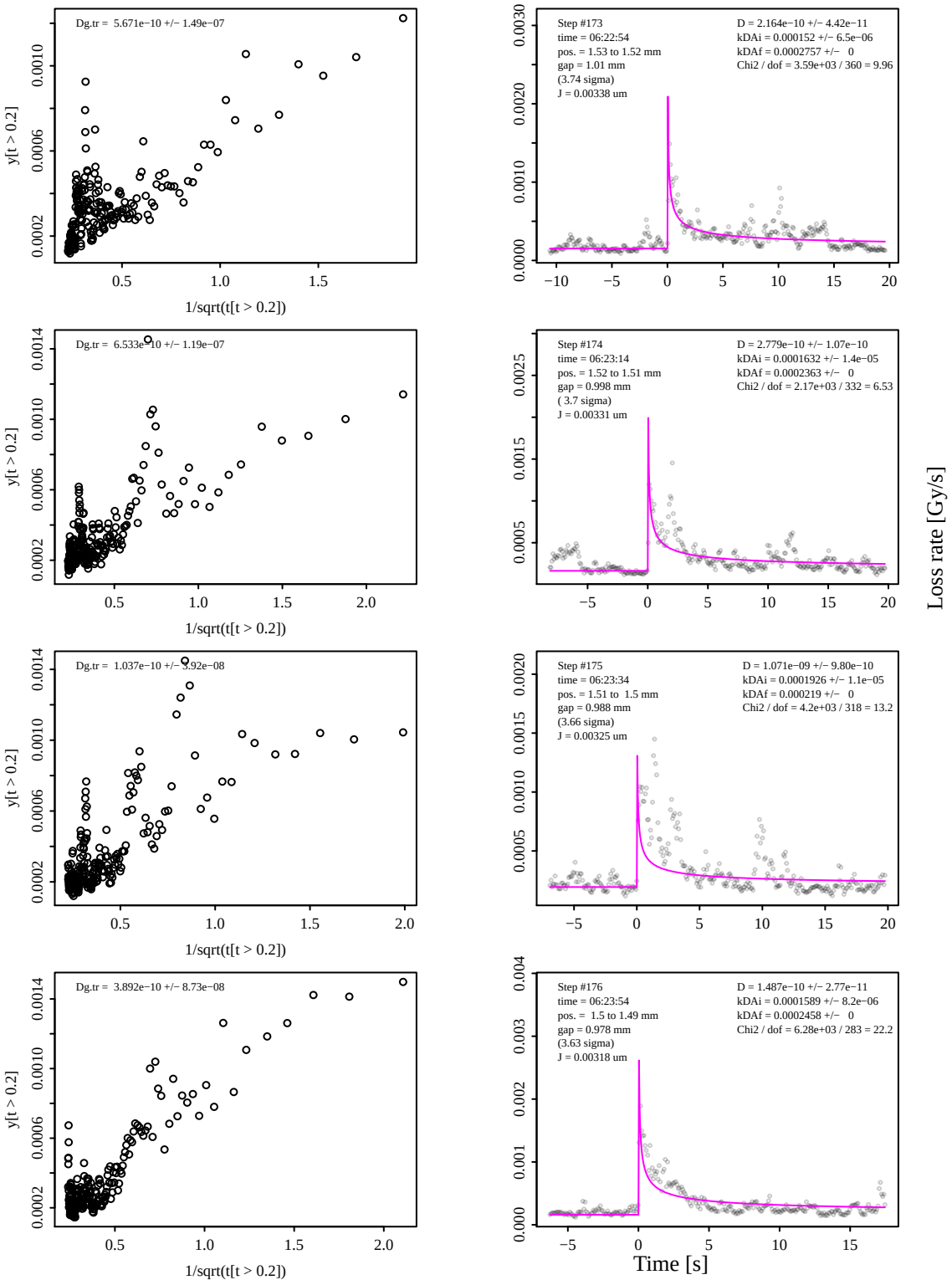

Dg.tr = 1.312e−10 +/− 1.11e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #153
time = 06:14:31
pos. = 1.75 to 1.74 mm
gap = 1.22 mm
(4.54 sigma)
J = 0.00498 um
D = 4.059e−10 +/− 1.47e−10
kDAi = 5.466e−05 +/− 1.2e−06
kDAf = 8.06e−05 +/− 0
Chi2 / dof = 795 / 239 = 3.33
Dg.tr = 8.444e−11 +/− 1.30e−08
Step #154
time = 06:14:46
pos. = 1.74 to 1.73 mm
gap = 1.21 mm
(4.5 sigma)
J = 0.0049 um
D = 1.567e−10 +/− 3.61e−11
kDAi = 6.857e−05 +/− 1.2e−06
kDAf = 0.0001074 +/− 0
Chi2 / dof = 2.5e+04 / 226 = 111
Dg.tr = 7.903e−11 +/− 9.32e−09
Step #155
time = 06:15:01
pos. = 1.73 to 1.72 mm
gap = 1.2 mm
(4.46 sigma)
J = 0.00482 um
D = 4.389e−10 +/− 1.80e−10
kDAi = 6.929e−05 +/− 6.5e−06
kDAf = 8.104e−05 +/− 0
Chi2 / dof = 1.71e+03 / 229 = 7.49
Dg.tr = 4.863e−11 +/− 9.35e−09
Step #156
time = 06:15:16
pos. = 1.72 to 1.71 mm
gap = 1.19 mm
(4.43 sigma)
J = 0.00474 um
D = 2.228e−10 +/− 7.47e−11
kDAi = 6.539e−05 +/− 4.8e−06
kDAf = 9.607e−05 +/− 0
Chi2 / dof = 1.49e+04 / 236 = 63.2
Loss rate [Gy/s]
Time [s]

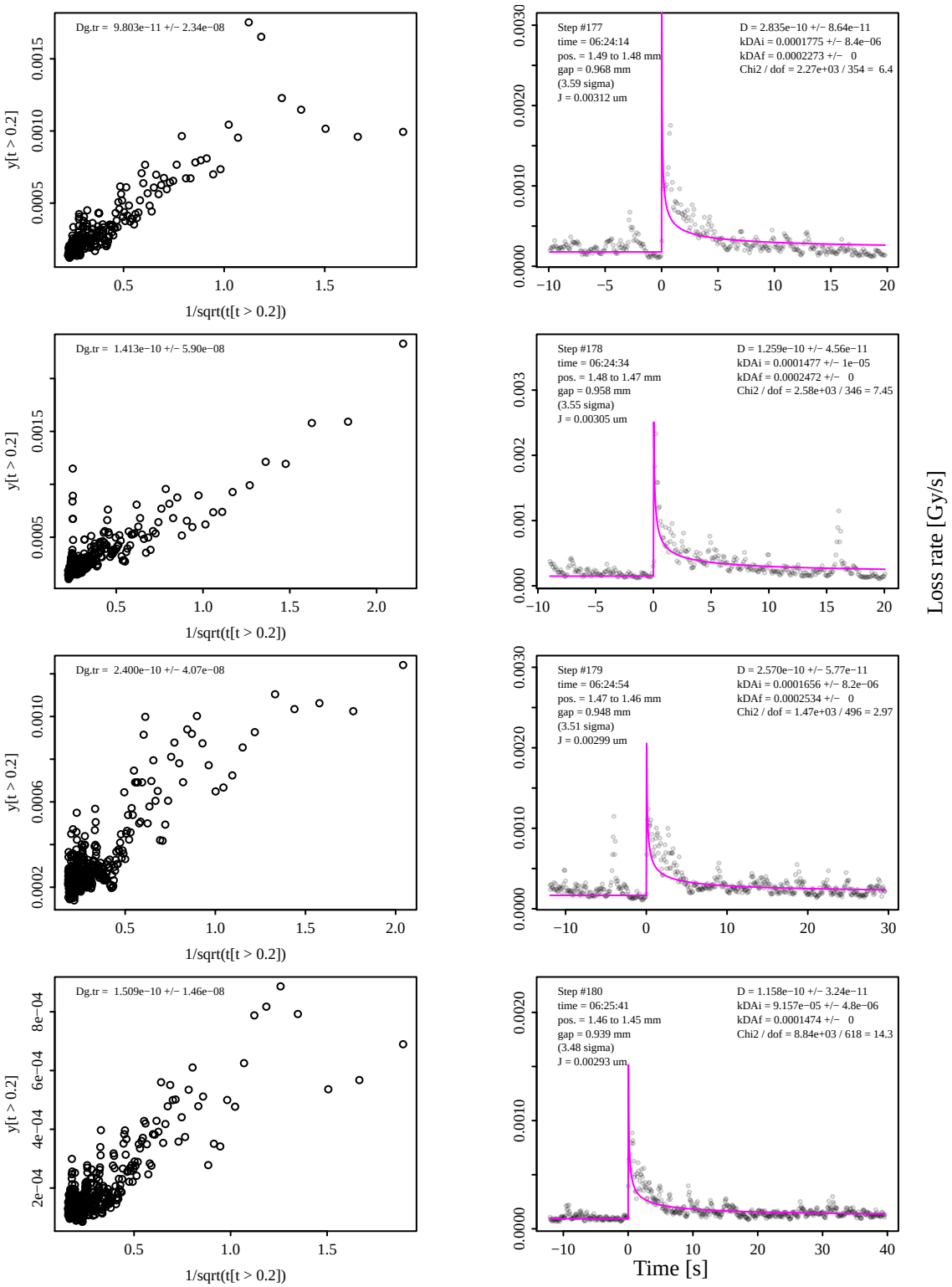

Dg.tr = 1.502e−10 +/− 2.27e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #157
time = 06:15:31
pos. = 1.71 to 1.7 mm
gap = 1.18 mm
(4.39 sigma)
J = 0.00466 um
D = 3.044e−10 +/− 3.95e−11
kDAi = 7.764e−05 +/− 3.6e−06
kDAf = 0.0001181 +/− 0
Chi2 / dof = 2.06e+03 / 263 = 7.85
Dg.tr = 4.097e−10 +/− 2.38e−08
Step #158
time = 06:15:46
pos. = 1.7 to 1.69 mm
gap = 1.17 mm
(4.35 sigma)
J = 0.00458 um
D = 3.593e−10 +/− 2.03e−10
kDAi = 7.852e−05 +/− 7.7e−06
kDAf = 0.000123 +/− 0
Chi2 / dof = 2.91e+03 / 234 = 12.5
Dg.tr = 4.778e−11 +/− 2.14e−09
Step #159
time = 06:16:01
pos. = 1.69 to 1.68 mm
gap = 1.16 mm
(4.31 sigma)
J = 0.0045 um
D = 4.298e−11 +/− 2.76e−12
kDAi = 3.64e−05 +/− 0
kDAf = 6.668e−05 +/− 3.3e−13
Chi2 / dof = 7.9e+03 / 894 = 8.83
Dg.tr = 3.349e−09 +/− 9.28e−08
Step #160
time = 06:17:11
pos. = 1.68 to 1.67 mm
gap = 1.15 mm
(4.28 sigma)
J = 0.00443 um
D = 1.749e−10 +/− 7.99e−12
kDAi = 3.791e−05 +/− 2.6e−07
kDAf = 8.166e−05 +/− 0
Chi2 / dof = 1.77e+04 / 303 = 58.3
Loss rate [Gy/s]
Time [s]

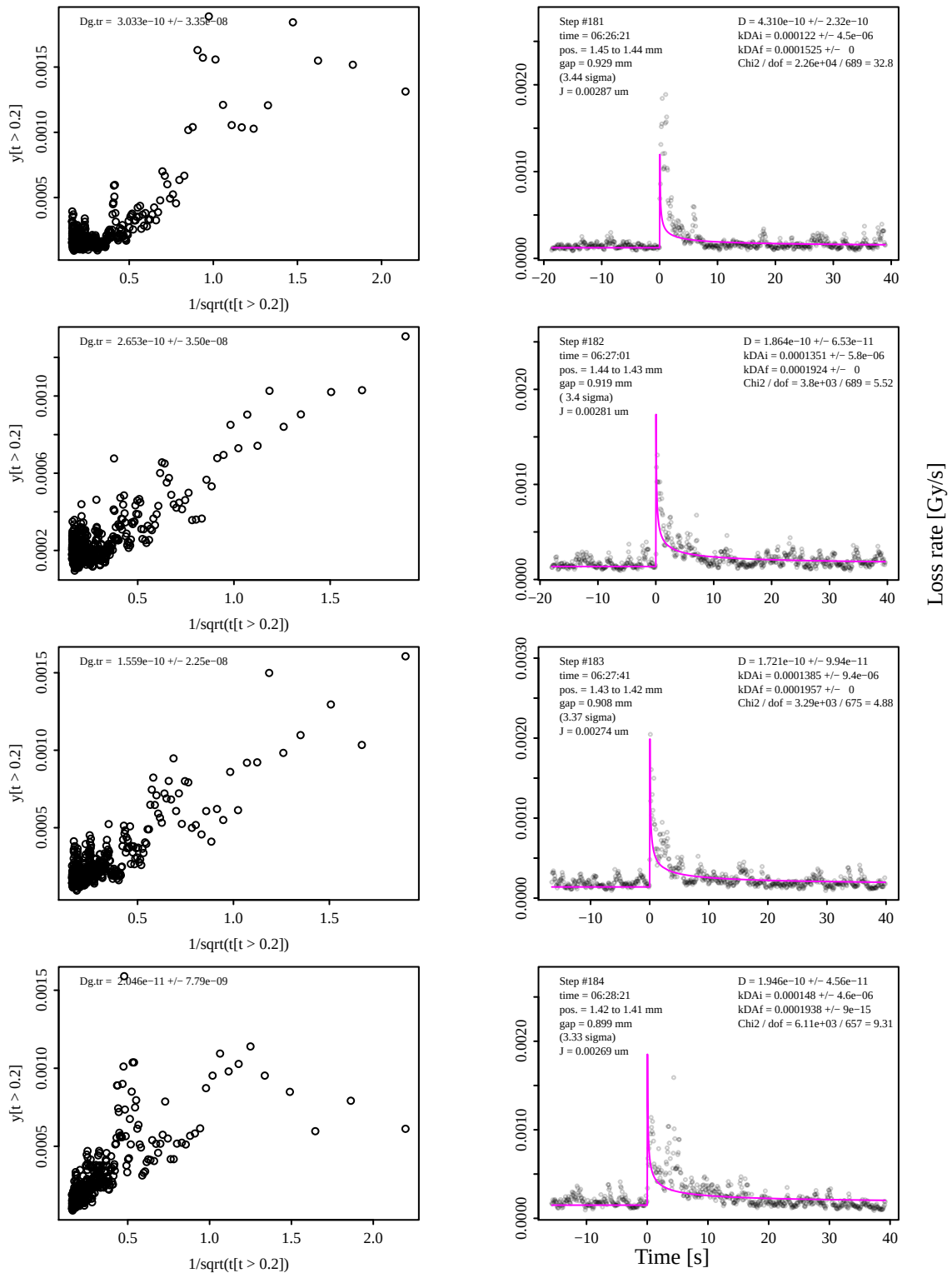

Dg.tr = 3.890e−10 +/− 4.63e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #161
time = 06:17:26
pos. = 1.67 to 1.66 mm
gap = 1.14 mm
(4.24 sigma)
J = 0.00435 um
D = 2.507e−10 +/− 3.38e−11
kDAi = 6.99e−05 +/− 1.2e−06
kDAf = 0.0001049 +/− 0
Chi2 / dof = 4.32e+03 / 256 = 16.9
Dg.tr = 2.441e−10 +/− 2.70e−08
Step #162
time = 06:17:41
pos. = 1.66 to 1.65 mm
gap = 1.13 mm
(4.2 sigma)
J = 0.00427 um
D = 2.625e−10 +/− 4.75e−11
kDAi = 8.778e−05 +/− 2.5e−06
kDAf = 0.0001319 +/− 0
Chi2 / dof = 3.3e+03 / 258 = 12.8
Dg.tr = 8.140e−11 +/− 1.78e−08
Step #163
time = 06:17:56
pos. = 1.65 to 1.64 mm
gap = 1.12 mm
(4.16 sigma)
J = 0.0042 um
D = 1.501e−10 +/− 3.07e−11
kDAi = 9.126e−05 +/− 8.4e−07
kDAf = 0.0001634 +/− 0
Chi2 / dof = 6.98e+03 / 250 = 27.9
Dg.tr = 1.298e−10 +/− 2.17e−08
Step #164
time = 06:18:10
pos. = 1.64 to 1.63 mm
gap = 1.11 mm
(4.13 sigma)
J = 0.00413 um
D = 5.566e−10 +/− 4.66e−10
kDAi = 0.0001214 +/− 1e−05
kDAf = 0.000177 +/− 0
Chi2 / dof = 7.46e+03 / 240 = 31.1
Loss rate [Gy/s]
Time [s]

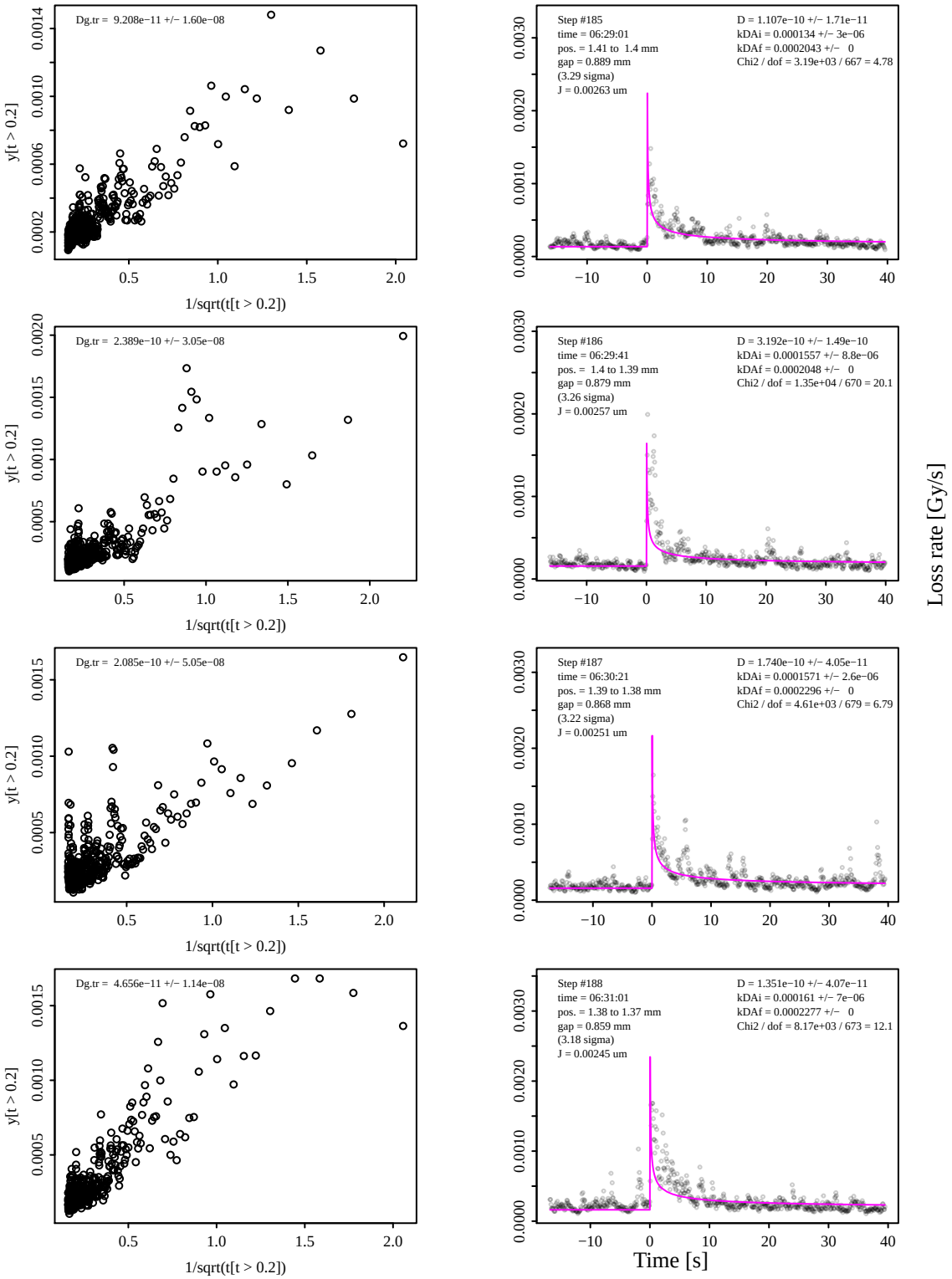

Dg.tr = 7.306e−11 +/− 1.83e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #165
time = 06:18:25
pos. = 1.63 to 1.62 mm
gap = 1.1 mm
(4.09 sigma)
J = 0.00405 um
D = 3.709e−09 +/− 6.65e−10
kDAi = 0.0001494 +/− 2.8e−06
kDAf = 0.0001618 +/− 0
Chi2 / dof = 3.47e+03 / 264 = 13.1
Dg.tr = 9.136e−11 +/− 1.70e−08
Step #166
time = 06:18:41
pos. = 1.62 to 1.61 mm
gap = 1.09 mm
(4.05 sigma)
J = 0.00398 um
D = 3.173e−10 +/− 1.72e−10
kDAi = 0.0001291 +/− 9.3e−06
kDAf = 0.0001886 +/− 0
Chi2 / dof = 1.68e+04 / 258 = 65
Dg.tr = 6.437e−11 +/− 2.60e−08
Step #167
time = 06:18:56
pos. = 1.61 to 1.6 mm
gap = 1.08 mm
(4.02 sigma)
J = 0.00391 um
D = 2.668e−10 +/− 6.44e−11
kDAi = 0.0001383 +/− 6.6e−06
kDAf = 0.0001791 +/− 0
Chi2 / dof = 5.34e+03 / 245 = 21.8
Dg.tr = 1.742e−10 +/− 5.43e−08
Step #168
time = 06:19:11
pos. = 1.6 to 1.59 mm
gap = 1.07 mm
(3.98 sigma)
J = 0.00384 um
D = 3.163e−10 +/− 4.45e−11
kDAi = 0.0001399 +/− 4.7e−06
kDAf = 0.0001878 +/− 0
Chi2 / dof = 5.38e+03 / 266 = 20.2
Loss rate [Gy/s]
Time [s]

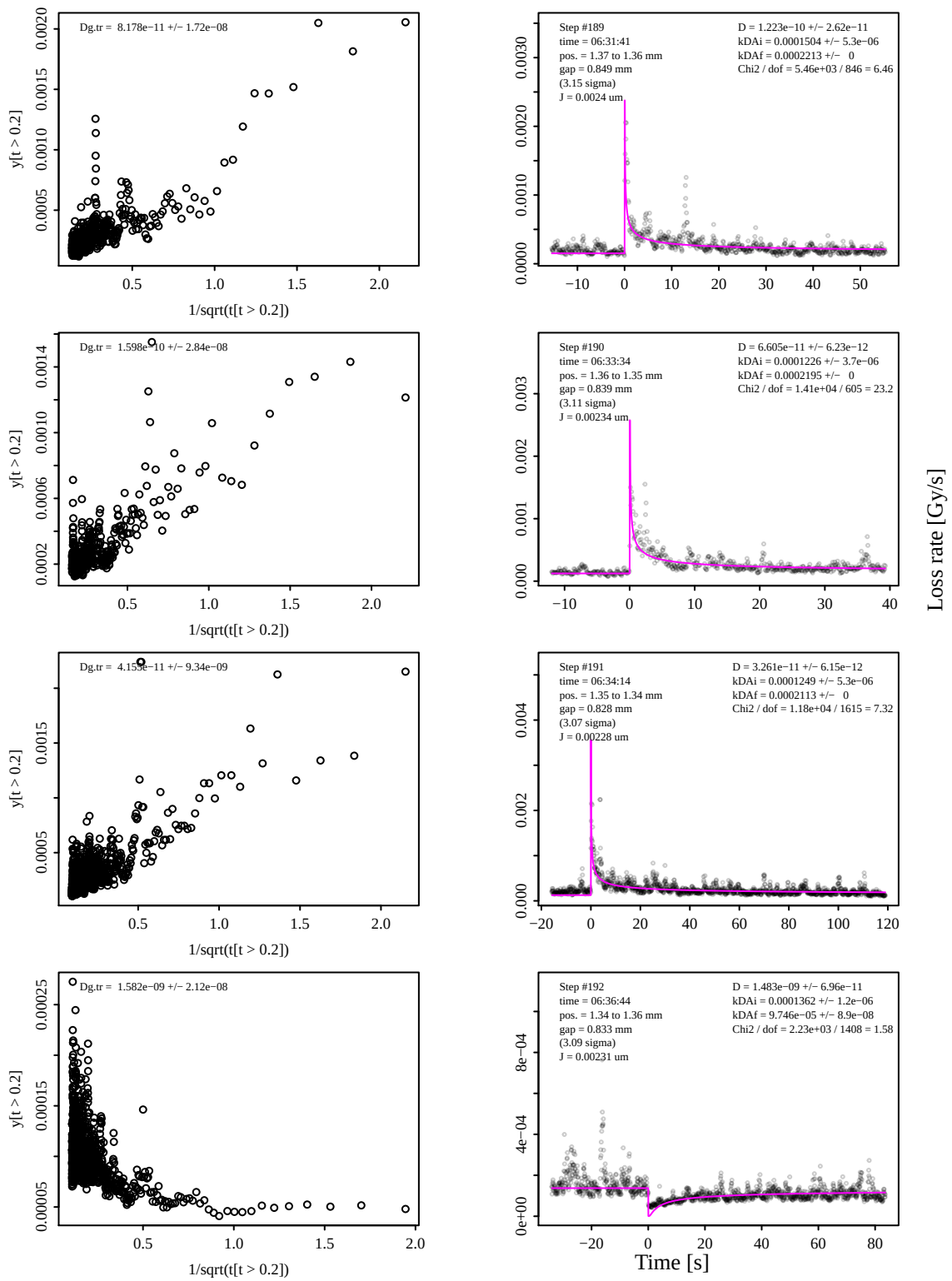

Dg.tr = 4.124e−11 +/− 1.37e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #169
time = 06:19:26
pos. = 1.59 to 1.58 mm
gap = 1.06 mm
(3.94 sigma)
J = 0.00376 um
D = 1.393e−10 +/− 4.09e−11
kDAi = 0.0001381 +/− 2.9e−06
kDAf = 0.0002036 +/− 0
Chi2 / dof = 5.53e+03 / 245 = 22.6
Dg.tr = 6.518e−11 +/− 1.12e−08
Step #170
time = 06:19:41
pos. = 1.58 to 1.57 mm
gap = 1.05 mm
(3.91 sigma)
J = 0.00369 um
D = 2.253e−10 +/− 1.80e−10
kDAi = 0.0001198 +/− 2.1e−05
kDAf = 0.0001732 +/− 4e−15
Chi2 / dof = 5.77e+03 / 348 = 16.6
Dg.tr = 7.056e−11 +/− 1.41e−08
Step #171
time = 06:22:14
pos. = 1.55 to 1.54 mm
gap = 1.03 mm
(3.81 sigma)
J = 0.00351 um
D = 5.218e−10 +/− 1.49e−10
kDAi = 0.0001119 +/− 1.6e−06
kDAf = 0.0001269 +/− 0
Chi2 / dof = 7.82e+03 / 487 = 16.1
Dg.tr = 8.030e−10 +/− 1.11e−07
Step #172
time = 06:22:34
pos. = 1.54 to 1.53 mm
gap = 1.02 mm
(3.77 sigma)
J = 0.00344 um
D = 4.889e−10 +/− 1.53e−10
kDAi = 0.0001148 +/− 1.7e−06
kDAf = 0.0001563 +/− 0
Chi2 / dof = 1.26e+04 / 335 = 37.7
Loss rate [Gy/s]
Time [s]

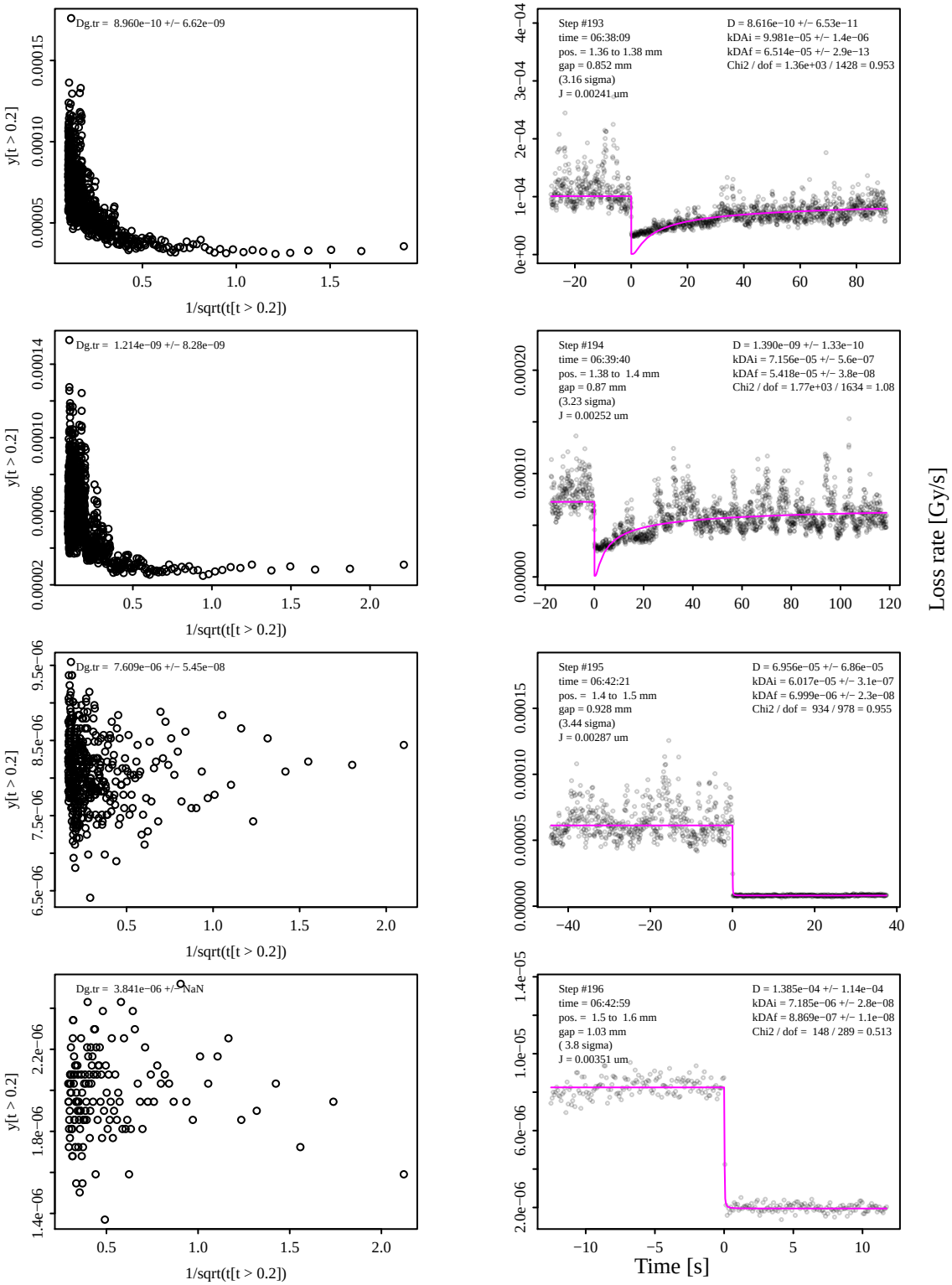

Dg.tr = 5.671e−10 +/− 1.49e−07
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #173
time = 06:22:54
pos. = 1.53 to 1.52 mm
gap = 1.01 mm
(3.74 sigma)
J = 0.00338 um
D = 2.164e−10 +/− 4.42e−11
kDAi = 0.000152 +/− 6.5e−06
kDAf = 0.0002757 +/− 0
Chi2 / dof = 3.59e+03 / 360 = 9.96
Dg.tr = 6.533e−10 +/− 1.19e−07
Step #174
time = 06:23:14
pos. = 1.52 to 1.51 mm
gap = 0.998 mm
(3.7 sigma)
J = 0.00331 um
D = 2.779e−10 +/− 1.07e−10
kDAi = 0.0001632 +/− 1.4e−05
kDAf = 0.0002363 +/− 0
Chi2 / dof = 2.17e+03 / 332 = 6.53
Dg.tr = 1.037e−10 +/− 3.92e−08
Step #175
time = 06:23:34
pos. = 1.51 to 1.5 mm
gap = 0.988 mm
(3.66 sigma)
J = 0.00325 um
D = 1.071e−09 +/− 9.80e−10
kDAi = 0.0001926 +/− 1.1e−05
kDAf = 0.000219 +/− 0
Chi2 / dof = 4.2e+03 / 318 = 13.2
Dg.tr = 3.892e−10 +/− 8.73e−08
Step #176
time = 06:23:54
pos. = 1.5 to 1.49 mm
gap = 0.978 mm
(3.63 sigma)
J = 0.00318 um
D = 1.487e−10 +/− 2.77e−11
kDAi = 0.0001589 +/− 8.2e−06
kDAf = 0.0002458 +/− 0
Chi2 / dof = 6.28e+03 / 283 = 22.2
Loss rate [Gy/s]
Time [s]

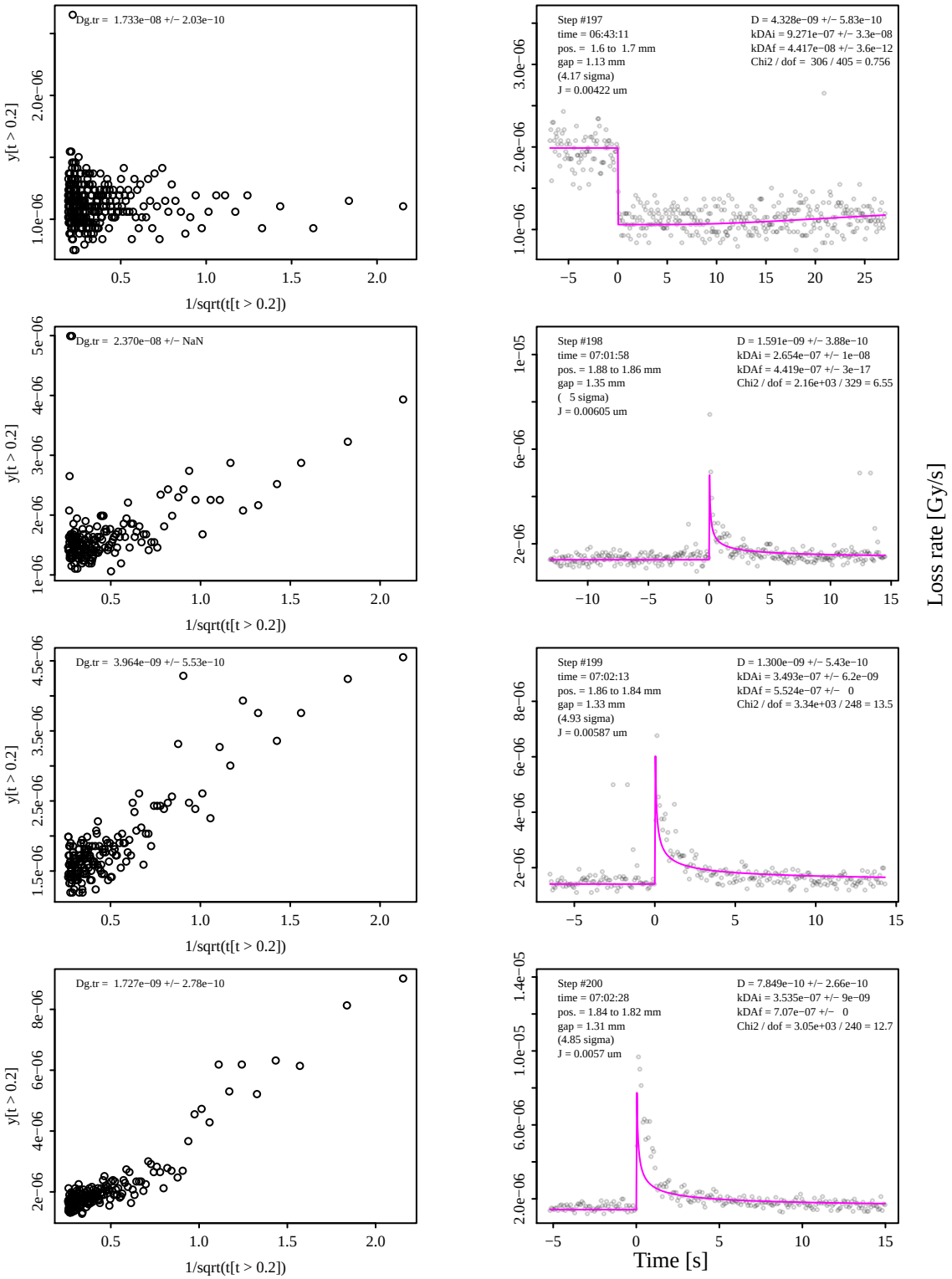

Dg.tr = 9.803e−11 +/− 2.34e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #177
time = 06:24:14
pos. = 1.49 to 1.48 mm
gap = 0.968 mm
(3.59 sigma)
J = 0.00312 um
D = 2.835e−10 +/− 8.64e−11
kDAi = 0.0001775 +/− 8.4e−06
kDAf = 0.0002273 +/− 0
Chi2 / dof = 2.27e+03 / 354 = 6.4
Dg.tr = 1.413e−10 +/− 5.90e−08
Step #178
time = 06:24:34
pos. = 1.48 to 1.47 mm
gap = 0.958 mm
(3.55 sigma)
J = 0.00305 um
D = 1.259e−10 +/− 4.56e−11
kDAi = 0.0001477 +/− 1e−05
kDAf = 0.0002472 +/− 0
Chi2 / dof = 2.58e+03 / 346 = 7.45
Dg.tr = 2.400e−10 +/− 4.07e−08
Step #179
time = 06:24:54
pos. = 1.47 to 1.46 mm
gap = 0.948 mm
(3.51 sigma)
J = 0.00299 um
D = 2.570e−10 +/− 5.77e−11
kDAi = 0.0001656 +/− 8.2e−06
kDAf = 0.0002534 +/− 0
Chi2 / dof = 1.47e+03 / 496 = 2.97
Dg.tr = 1.509e−10 +/− 1.46e−08
Step #180
time = 06:25:41
pos. = 1.46 to 1.45 mm
gap = 0.939 mm
(3.48 sigma)
J = 0.00293 um
D = 1.158e−10 +/− 3.24e−11
kDAi = 9.157e−05 +/− 4.8e−06
kDAf = 0.0001474 +/− 0
Chi2 / dof = 8.84e+03 / 618 = 14.3
Loss rate [Gy/s]
Time [s]

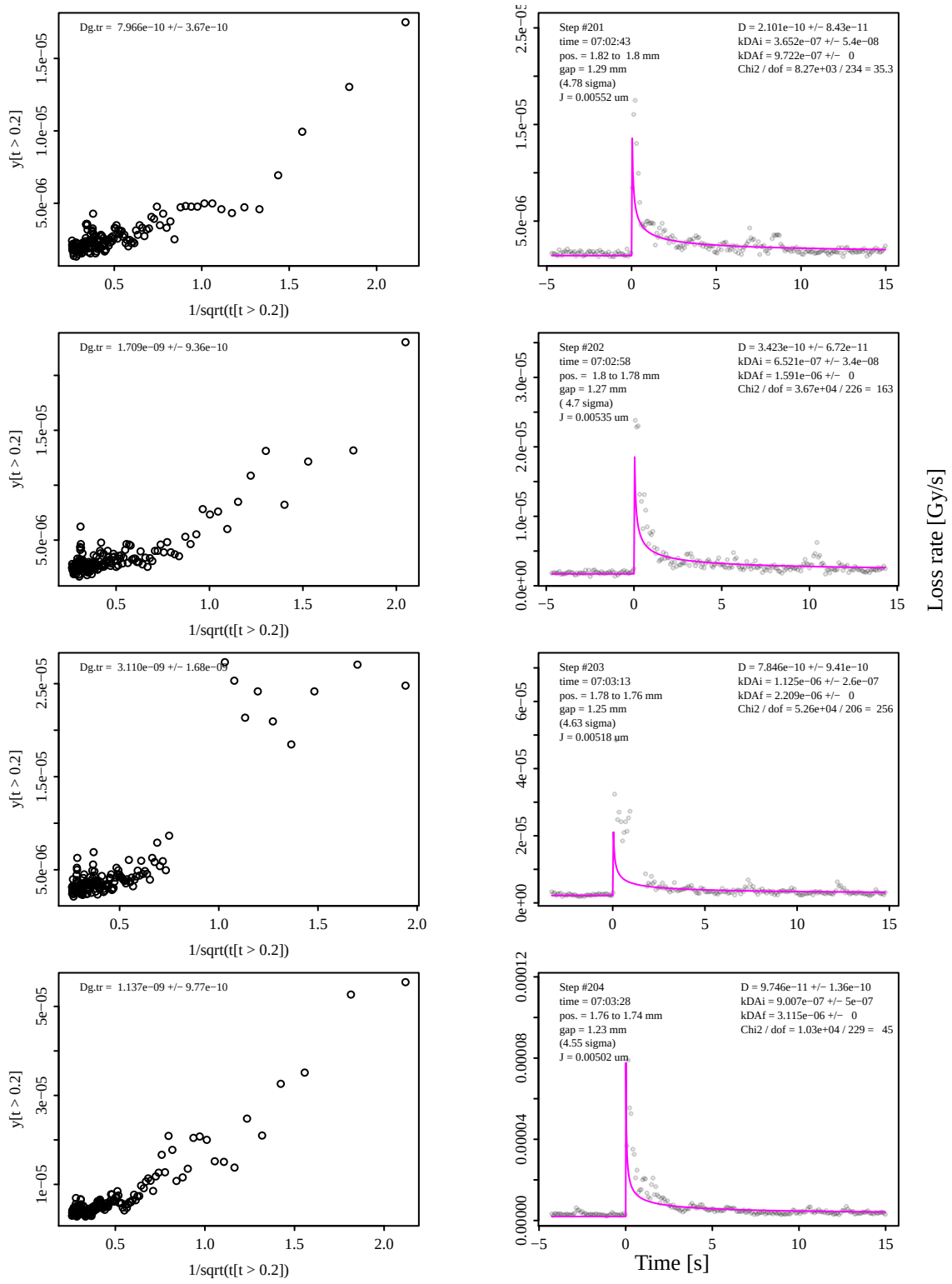

Dg.tr = 3.033e−10 +/− 3.35e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #181
time = 06:26:21
pos. = 1.45 to 1.44 mm
gap = 0.929 mm
(3.44 sigma)
J = 0.00287 um
D = 4.310e−10 +/− 2.32e−10
kDAi = 0.000122 +/− 4.5e−06
kDAf = 0.0001525 +/− 0
Chi2 / dof = 2.26e+04 / 689 = 32.8
Dg.tr = 2.653e−10 +/− 3.50e−08
Step #182
time = 06:27:01
pos. = 1.44 to 1.43 mm
gap = 0.919 mm
(3.4 sigma)
J = 0.00281 um
D = 1.864e−10 +/− 6.53e−11
kDAi = 0.0001351 +/− 5.8e−06
kDAf = 0.0001924 +/− 0
Chi2 / dof = 3.8e+03 / 689 = 5.52
Dg.tr = 1.559e−10 +/− 2.25e−08
Step #183
time = 06:27:41
pos. = 1.43 to 1.42 mm
gap = 0.908 mm
(3.37 sigma)
J = 0.00274 um
D = 1.721e−10 +/− 9.94e−11
kDAi = 0.0001385 +/− 9.4e−06
kDAf = 0.0001957 +/− 0
Chi2 / dof = 3.29e+03 / 675 = 4.88
Dg.tr = 2.046e−11 +/− 7.79e−09
Step #184
time = 06:28:21
pos. = 1.42 to 1.41 mm
gap = 0.899 mm
(3.33 sigma)
J = 0.00269 um
D = 1.946e−10 +/− 4.56e−11
kDAi = 0.000148 +/− 4.6e−06
kDAf = 0.0001938 +/− 9e−15
Chi2 / dof = 6.11e+03 / 657 = 9.31
Loss rate [Gy/s]
Time [s]

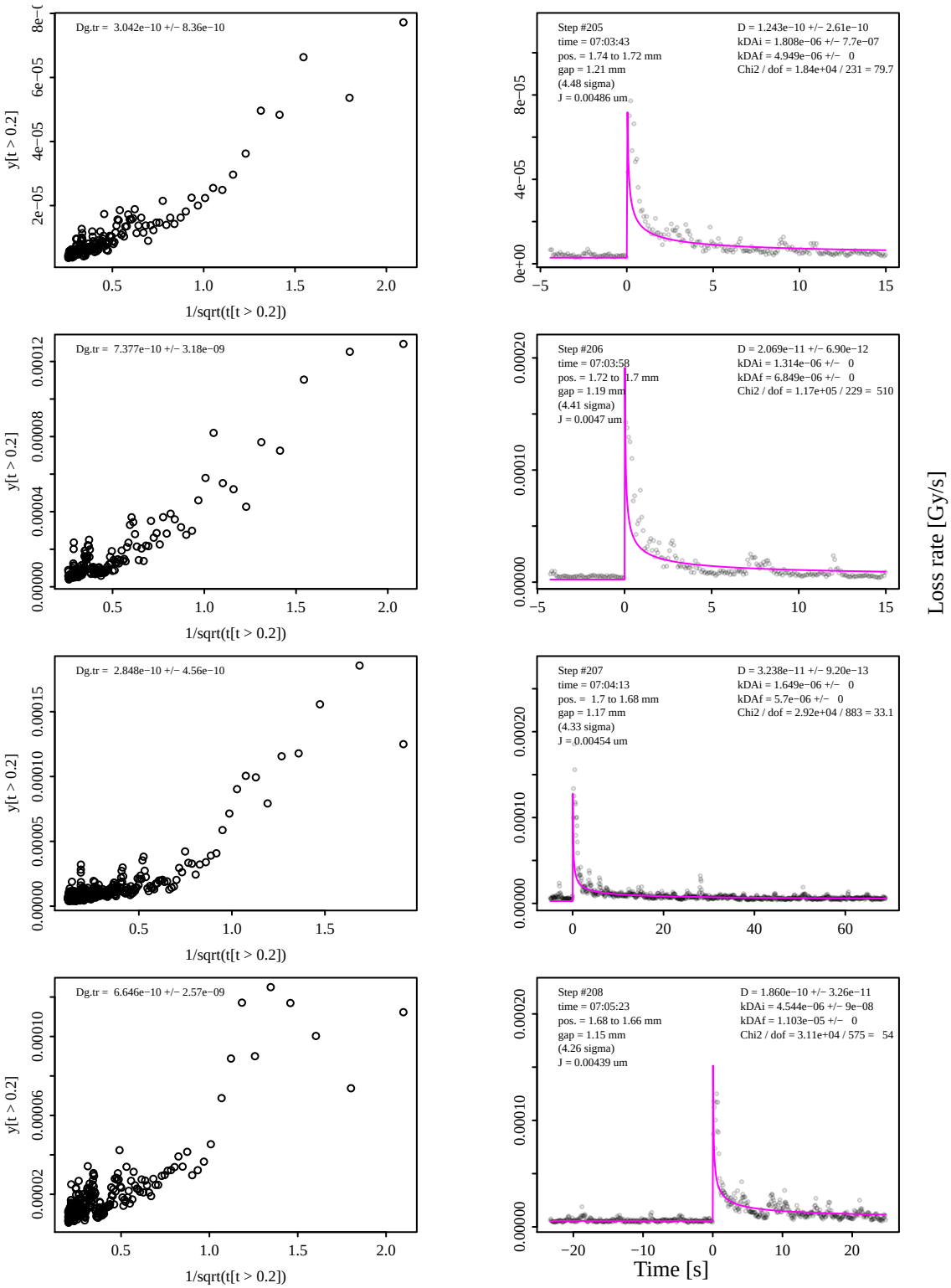

Dg.tr = 9.208e−11 +/− 1.60e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #185
time = 06:29:01
pos. = 1.41 to 1.4 mm
gap = 0.889 mm
(3.29 sigma)
J = 0.00263 um
D = 1.107e−10 +/− 1.71e−11
kDAi = 0.000134 +/− 3e−06
kDAf = 0.0002043 +/− 0
Chi2 / dof = 3.19e+03 / 667 = 4.78
Dg.tr = 2.389e−10 +/− 3.05e−08
Step #186
time = 06:29:41
pos. = 1.4 to 1.39 mm
gap = 0.879 mm
(3.26 sigma)
J = 0.00257 um
D = 3.192e−10 +/− 1.49e−10
kDAi = 0.0001557 +/− 8.8e−06
kDAf = 0.0002048 +/− 0
Chi2 / dof = 1.35e+04 / 670 = 20.1
Dg.tr = 2.085e−10 +/− 5.05e−08
Step #187
time = 06:30:21
pos. = 1.39 to 1.38 mm
gap = 0.868 mm
(3.22 sigma)
J = 0.00251 um
D = 1.740e−10 +/− 4.05e−11
kDAi = 0.0001571 +/− 2.6e−06
kDAf = 0.0002296 +/− 0
Chi2 / dof = 4.61e+03 / 679 = 6.79
Dg.tr = 4.656e−11 +/− 1.14e−08
Step #188
time = 06:31:01
pos. = 1.38 to 1.37 mm
gap = 0.859 mm
(3.18 sigma)
J = 0.00245 um
D = 1.351e−10 +/− 4.07e−11
kDAi = 0.000161 +/− 7e−06
kDAf = 0.0002277 +/− 0
Chi2 / dof = 8.17e+03 / 673 = 12.1
Loss rate [Gy/s]
Time [s]

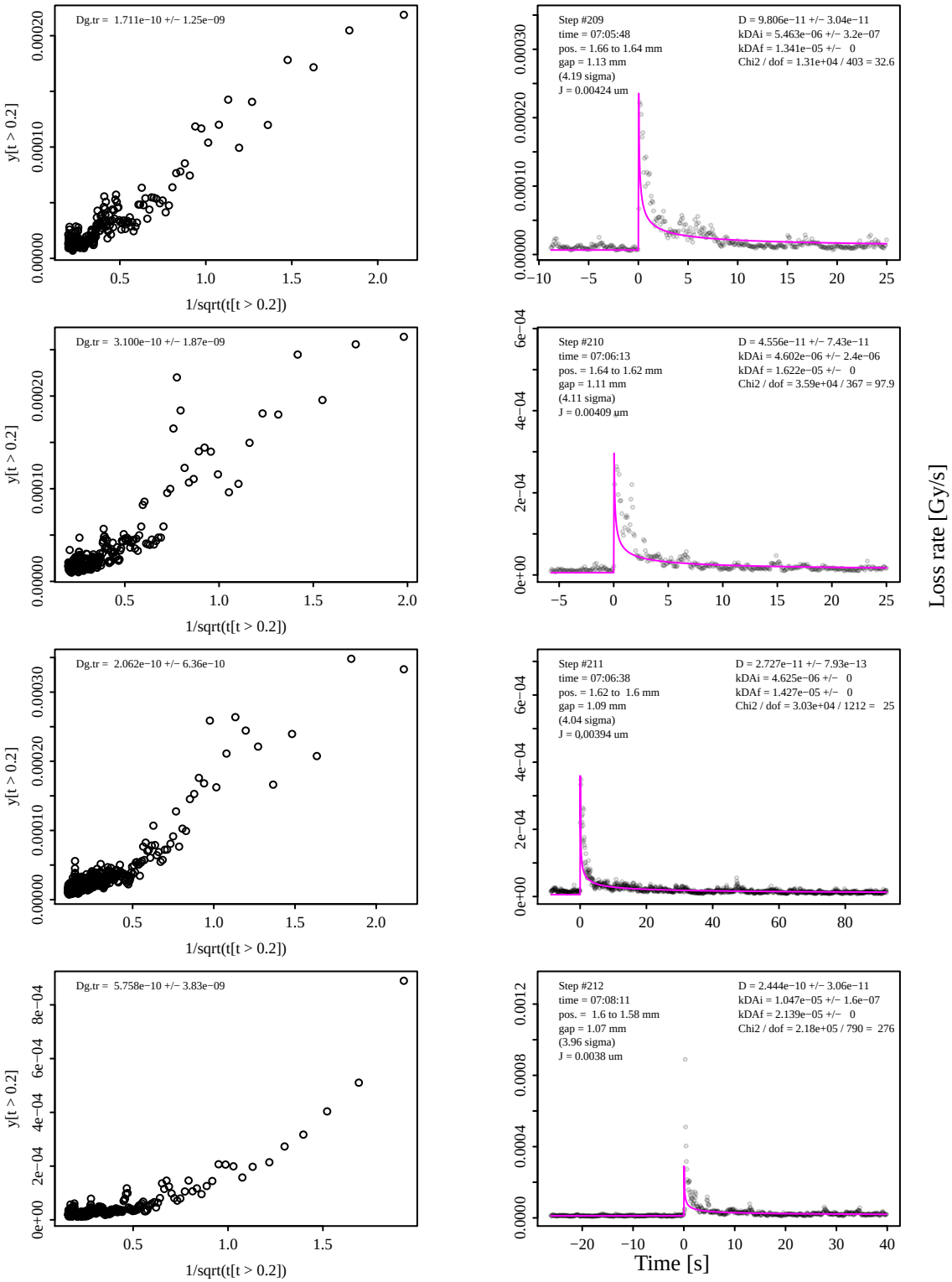

Dg.tr = 8.178e−11 +/− 1.72e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #189
time = 06:31:41
pos. = 1.37 to 1.36 mm
gap = 0.849 mm
(3.15 sigma)
J = 0.0024 um
D = 1.223e−10 +/− 2.62e−11
kDAi = 0.0001504 +/− 5.3e−06
kDAf = 0.0002213 +/− 0
Chi2 / dof = 5.46e+03 / 846 = 6.46
Dg.tr = 1.598e−10 +/− 2.84e−08
Step #190
time = 06:33:34
pos. = 1.36 to 1.35 mm
gap = 0.839 mm
(3.11 sigma)
J = 0.00234 um
D = 6.605e−11 +/− 6.23e−12
kDAi = 0.0001226 +/− 3.7e−06
kDAf = 0.0002195 +/− 0
Chi2 / dof = 1.41e+04 / 605 = 23.2
Dg.tr = 4.153e−11 +/− 9.34e−09
Step #191
time = 06:34:14
pos. = 1.35 to 1.34 mm
gap = 0.828 mm
(3.07 sigma)
J = 0.00228 um
D = 3.261e−11 +/− 6.15e−12
kDAi = 0.0001249 +/− 5.3e−06
kDAf = 0.0002113 +/− 0
Chi2 / dof = 1.18e+04 / 1615 = 7.32
Dg.tr = 1.582e−09 +/− 2.12e−08
Step #192
time = 06:36:44
pos. = 1.34 to 1.36 mm
gap = 0.833 mm
(3.09 sigma)
J = 0.00231 um
D = 1.483e−09 +/− 6.96e−11
kDAi = 0.0001362 +/− 1.2e−06
kDAf = 9.746e−05 +/− 8.9e−08
Chi2 / dof = 2.23e+03 / 1408 = 1.58
Loss rate [Gy/s]
Time [s]

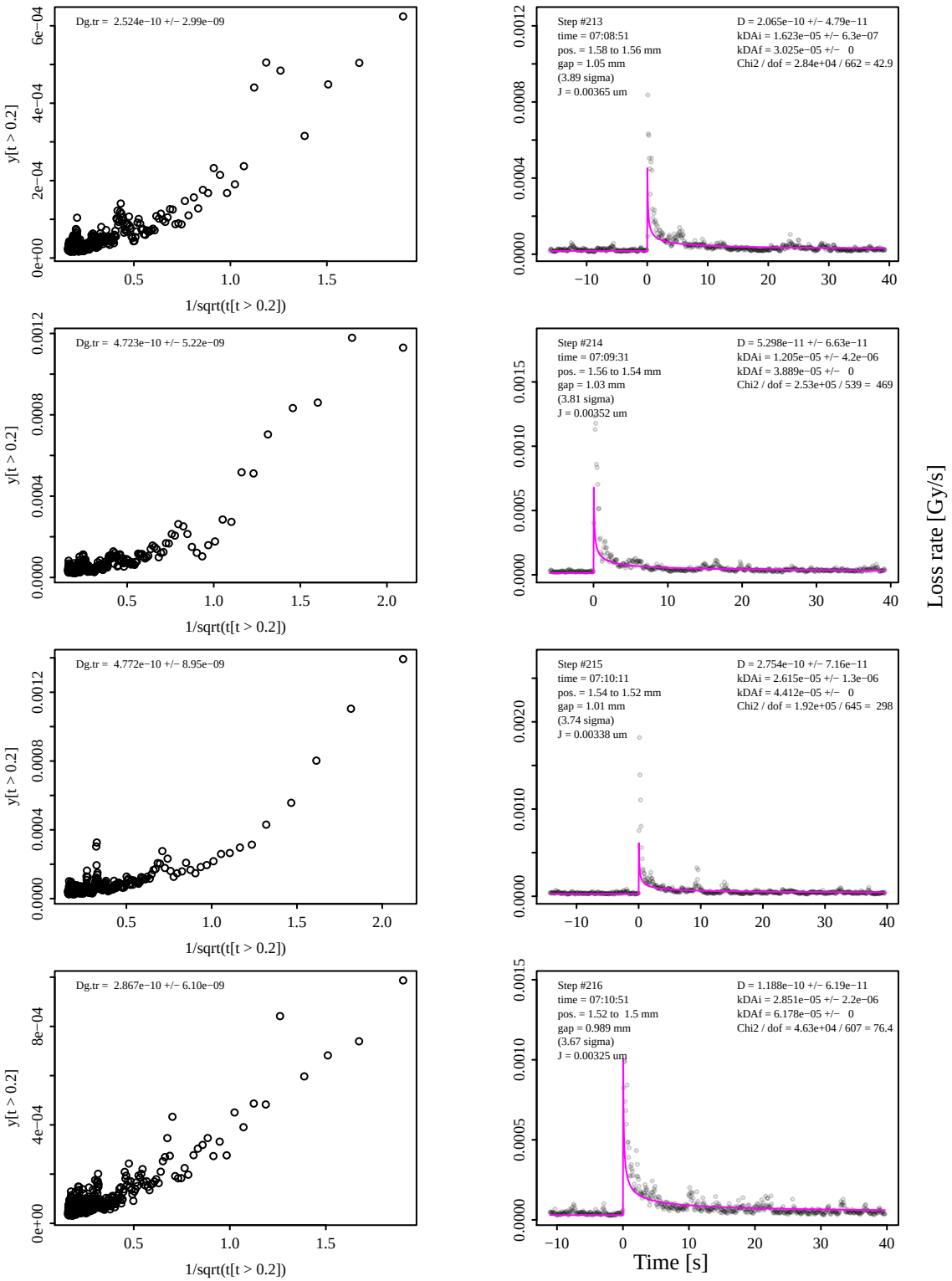

Dg.tr = 8.960e−10 +/− 6.62e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #193
time = 06:38:09
pos. = 1.36 to 1.38 mm
gap = 0.852 mm
(3.16 sigma)
J = 0.00241 um
D = 8.616e−10 +/− 6.53e−11
kDAi = 9.981e−05 +/− 1.4e−06
kDAf = 6.514e−05 +/− 2.9e−13
Chi2 / dof = 1.36e+03 / 1428 = 0.953
Dg.tr = 1.214e−09 +/− 8.28e−09
Step #194
time = 06:39:40
pos. = 1.38 to 1.4 mm
gap = 0.87 mm
(3.23 sigma)
J = 0.00252 um
D = 1.390e−09 +/− 1.33e−10
kDAi = 7.156e−05 +/− 5.6e−07
kDAf = 5.418e−05 +/− 3.8e−08
Chi2 / dof = 1.77e+03 / 1634 = 1.08
Dg.tr = 7.609e−06 +/− 5.45e−08
Step #195
time = 06:42:21
pos. = 1.4 to 1.5 mm
gap = 0.928 mm
(3.44 sigma)
J = 0.00287 um
D = 6.956e−05 +/− 6.86e−05
kDAi = 6.017e−05 +/− 3.1e−07
kDAf = 6.999e−06 +/− 2.3e−08
Chi2 / dof = 934 / 978 = 0.955
Dg.tr = 3.841e−06 +/− NaN
Step #196
time = 06:42:59
pos. = 1.5 to 1.6 mm
gap = 1.03 mm
(3.8 sigma)
J = 0.00351 um
D = 1.385e−04 +/− 1.14e−04
kDAi = 7.185e−06 +/− 2.8e−08
kDAf = 8.869e−07 +/− 1.1e−08
Chi2 / dof = 148 / 289 = 0.513
Loss rate [Gy/s]
Time [s]

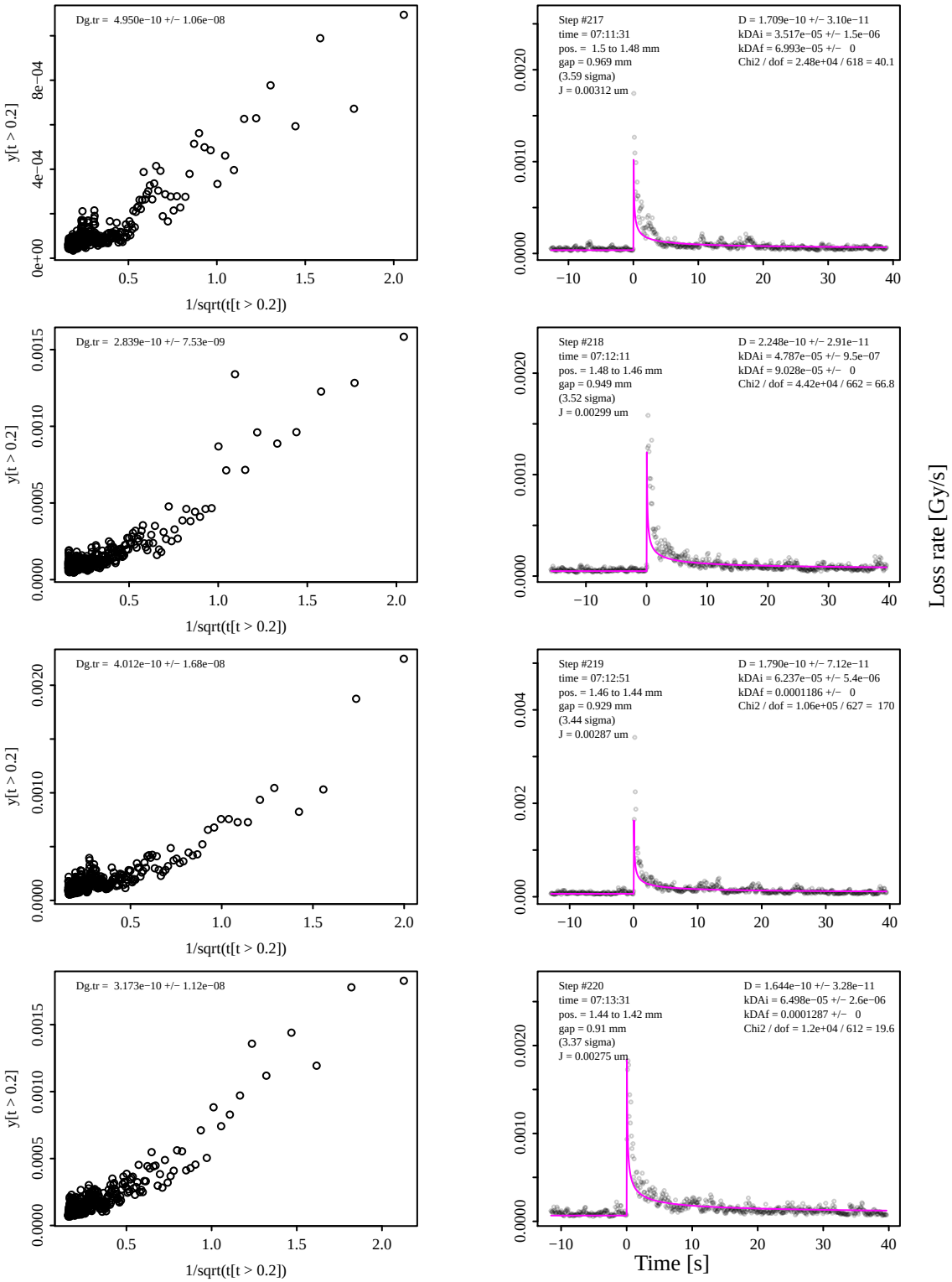

Dg.tr = 1.733e−08 +/− 2.03e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #197
time = 06:43:11
pos. = 1.6 to 1.7 mm
gap = 1.13 mm
(4.17 sigma)
J = 0.00422 um
D = 4.328e−09 +/− 5.83e−10
kDAi = 9.271e−07 +/− 3.3e−08
kDAf = 4.417e−08 +/− 3.6e−12
Chi2 / dof = 306 / 405 = 0.756
Dg.tr = 2.370e−08 +/− NaN
Step #198
time = 07:01:58
pos. = 1.88 to 1.86 mm
gap = 1.35 mm
(5 sigma)
J = 0.00605 um
D = 1.591e−09 +/− 3.88e−10
kDAi = 2.654e−07 +/− 1e−08
kDAf = 4.419e−07 +/− 3e−17
Chi2 / dof = 2.16e+03 / 329 = 6.55
Dg.tr = 3.964e−09 +/− 5.53e−10
Step #199
time = 07:02:13
pos. = 1.86 to 1.84 mm
gap = 1.33 mm
(4.93 sigma)
J = 0.00587 um
D = 1.300e−09 +/− 5.43e−10
kDAi = 3.493e−07 +/− 6.2e−09
kDAf = 5.524e−07 +/− 0
Chi2 / dof = 3.34e+03 / 248 = 13.5
Dg.tr = 1.727e−09 +/− 2.78e−10
Step #200
time = 07:02:28
pos. = 1.84 to 1.82 mm
gap = 1.31 mm
(4.85 sigma)
J = 0.0057 um
D = 7.849e−10 +/− 2.66e−10
kDAi = 3.535e−07 +/− 9e−09
kDAf = 7.07e−07 +/− 0
Chi2 / dof = 3.05e+03 / 240 = 12.7
Loss rate [Gy/s]
Time [s]

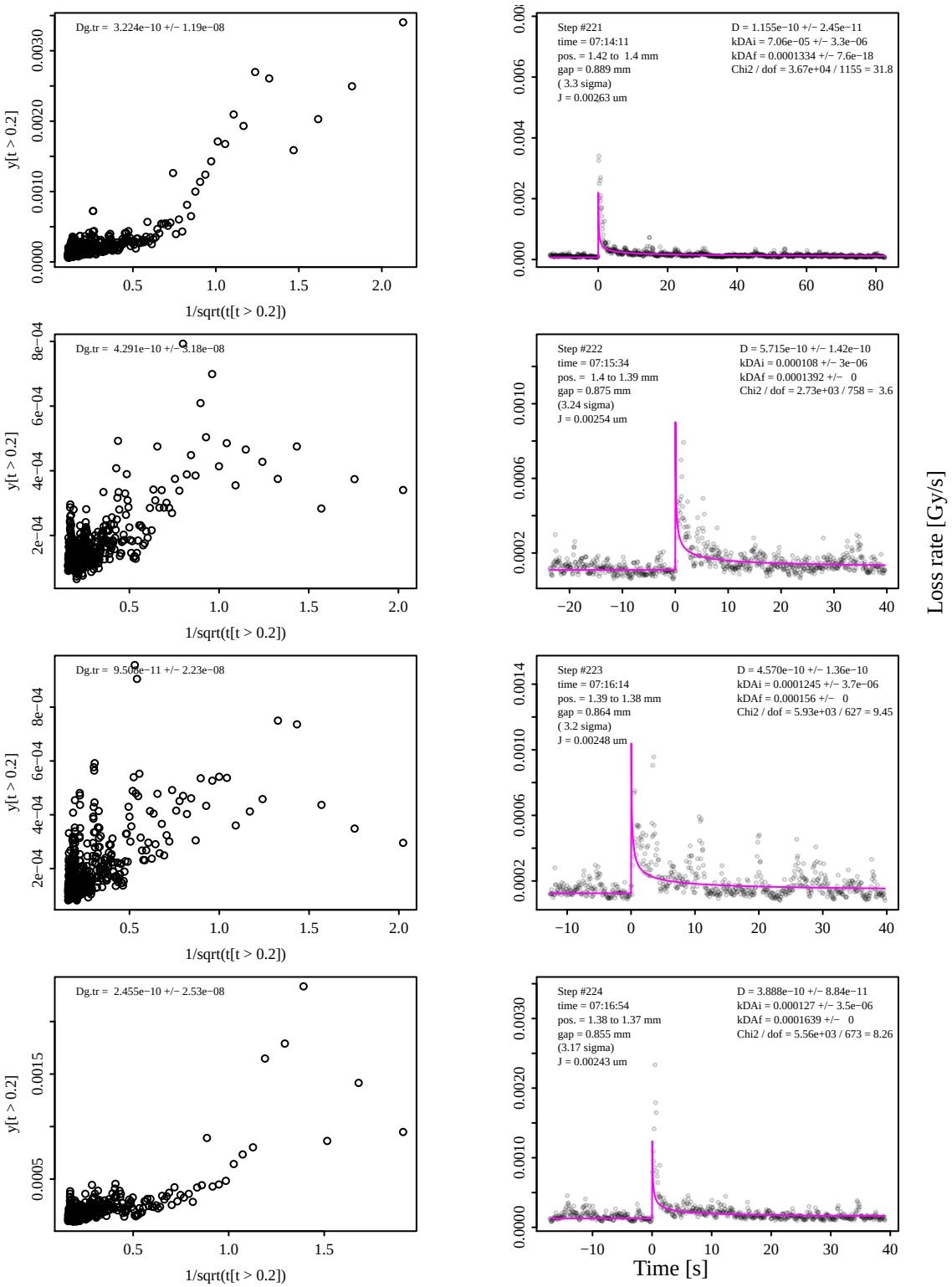

Dg.tr = 7.966e−10 +/− 3.67e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #201
time = 07:02:43
pos. = 1.82 to 1.8 mm
gap = 1.29 mm
(4.78 sigma)
J = 0.00552 um
D = 2.101e−10 +/− 8.43e−11
kDAi = 3.652e−07 +/− 5.4e−08
kDAf = 9.722e−07 +/− 0
Chi2 / dof = 8.27e+03 / 234 = 35.3
Dg.tr = 1.709e−09 +/− 9.36e−10
Step #202
time = 07:02:58
pos. = 1.8 to 1.78 mm
gap = 1.27 mm
(4.7 sigma)
J = 0.00535 um
D = 3.423e−10 +/− 6.72e−11
kDAi = 6.521e−07 +/− 3.4e−08
kDAf = 1.591e−06 +/− 0
Chi2 / dof = 3.67e+04 / 226 = 163
Dg.tr = 3.110e−09 +/− 1.68e−09
Step #203
time = 07:03:13
pos. = 1.78 to 1.76 mm
gap = 1.25 mm
(4.63 sigma)
J = 0.00518 um
D = 7.846e−10 +/− 9.41e−10
kDAi = 1.125e−06 +/− 2.6e−07
kDAf = 2.209e−06 +/− 0
Chi2 / dof = 5.26e+04 / 206 = 256
Dg.tr = 1.137e−09 +/− 9.77e−10
Step #204
time = 07:03:28
pos. = 1.76 to 1.74 mm
gap = 1.23 mm
(4.55 sigma)
J = 0.00502 um
D = 9.746e−11 +/− 1.36e−10
kDAi = 9.007e−07 +/− 5e−07
kDAf = 3.115e−06 +/− 0
Chi2 / dof = 1.03e+04 / 229 = 45
Loss rate [Gy/s]
Time [s]

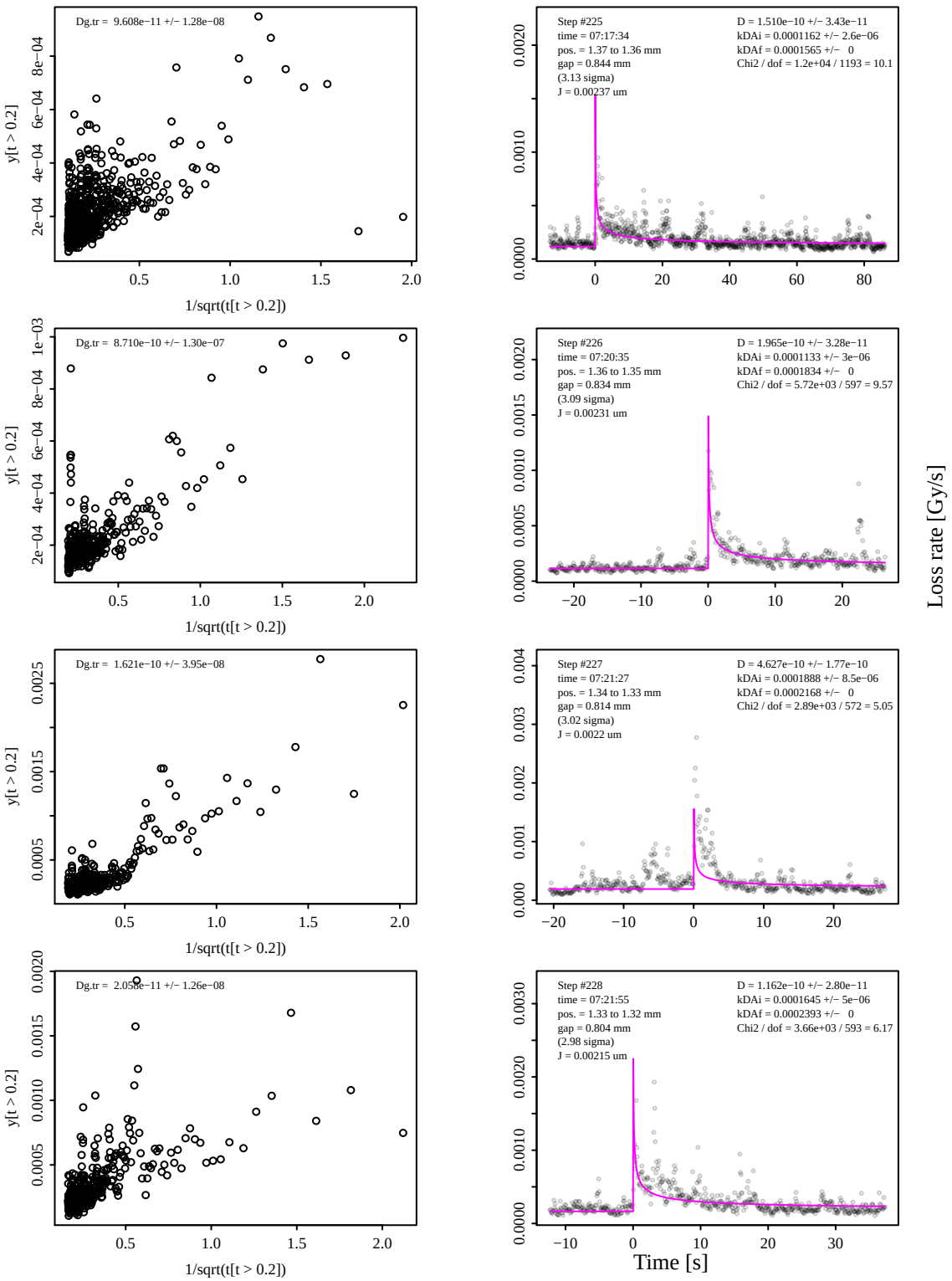

Dg.tr = 3.042e−10 +/− 8.36e−10
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #205
time = 07:03:43
pos. = 1.74 to 1.72 mm
gap = 1.21 mm
(4.48 sigma)
J = 0.00486 um
D = 1.243e−10 +/− 2.61e−10
kDAi = 1.808e−06 +/− 7.7e−07
kDAf = 4.949e−06 +/− 0
Chi2 / dof = 1.84e+04 / 231 = 79.7
Dg.tr = 7.377e−10 +/− 3.18e−09
Step #206
time = 07:03:58
pos. = 1.72 to 1.7 mm
gap = 1.19 mm
(4.41 sigma)
J = 0.0047 um
D = 2.069e−11 +/− 6.90e−12
kDAi = 1.314e−06 +/− 0
kDAf = 6.849e−06 +/− 0
Chi2 / dof = 1.17e+05 / 229 = 510
Dg.tr = 2.848e−10 +/− 4.56e−10
Step #207
time = 07:04:13
pos. = 1.7 to 1.68 mm
gap = 1.17 mm
(4.33 sigma)
J = 0.00454 um
D = 3.238e−11 +/− 9.20e−13
kDAi = 1.649e−06 +/− 0
kDAf = 5.7e−06 +/− 0
Chi2 / dof = 2.92e+04 / 883 = 33.1
Dg.tr = 6.646e−10 +/− 2.57e−09
Step #208
time = 07:05:23
pos. = 1.68 to 1.66 mm
gap = 1.15 mm
(4.26 sigma)
J = 0.00439 um
D = 1.860e−10 +/− 3.26e−11
kDAi = 4.544e−06 +/− 9e−08
kDAf = 1.103e−05 +/− 0
Chi2 / dof = 3.11e+04 / 575 = 54
Loss rate [Gy/s]
Time [s]

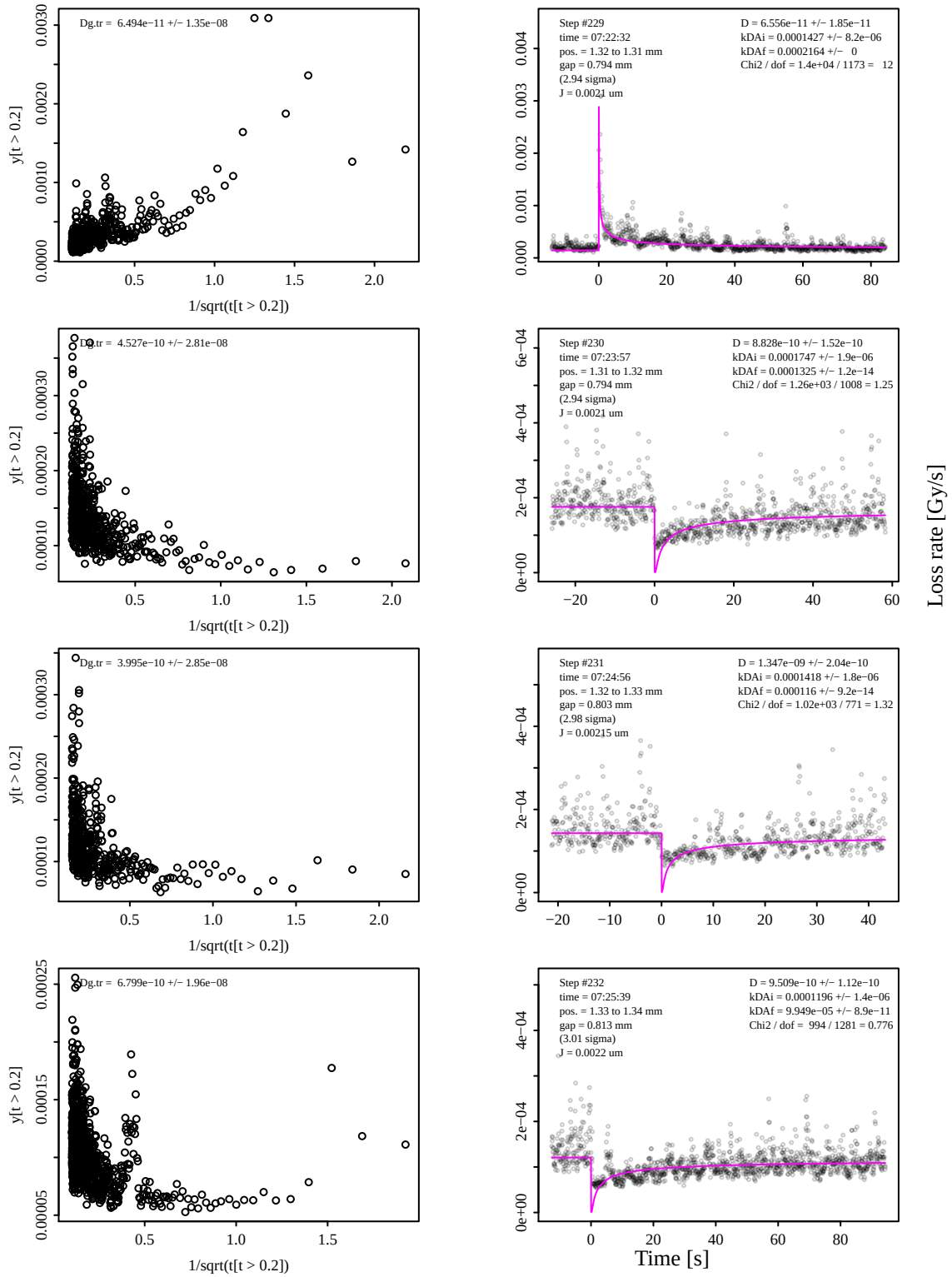

Dg.tr = 1.711e−10 +/− 1.25e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #209
time = 07:05:48
pos. = 1.66 to 1.64 mm
gap = 1.13 mm
(4.19 sigma)
J = 0.00424 um
D = 9.806e−11 +/− 3.04e−11
kDAi = 5.463e−06 +/− 3.2e−07
kDAf = 1.341e−05 +/− 0
Chi2 / dof = 1.31e+04 / 403 = 32.6
Dg.tr = 3.100e−10 +/− 1.87e−09
Step #210
time = 07:06:13
pos. = 1.64 to 1.62 mm
gap = 1.11 mm
(4.11 sigma)
J = 0.00409 um
D = 4.556e−11 +/− 7.43e−11
kDAi = 4.602e−06 +/− 2.4e−06
kDAf = 1.622e−05 +/− 0
Chi2 / dof = 3.59e+04 / 367 = 97.9
Dg.tr = 2.062e−10 +/− 6.36e−10
Step #211
time = 07:06:38
pos. = 1.62 to 1.6 mm
gap = 1.09 mm
(4.04 sigma)
J = 0,00394 um
D = 2.727e−11 +/− 7.93e−13
kDAi = 4.625e−06 +/− 0
kDAf = 1.427e−05 +/− 0
Chi2 / dof = 3.03e+04 / 1212 = 25
Dg.tr = 5.758e−10 +/− 3.83e−09
Step #212
time = 07:08:11
pos. = 1.6 to 1.58 mm
gap = 1.07 mm
(3.96 sigma)
J = 0.0038 um
D = 2.444e−10 +/− 3.06e−11
kDAi = 1.047e−05 +/− 1.6e−07
kDAf = 2.139e−05 +/− 0
Chi2 / dof = 2.18e+05 / 790 = 276
Loss rate [Gy/s]
Time [s]

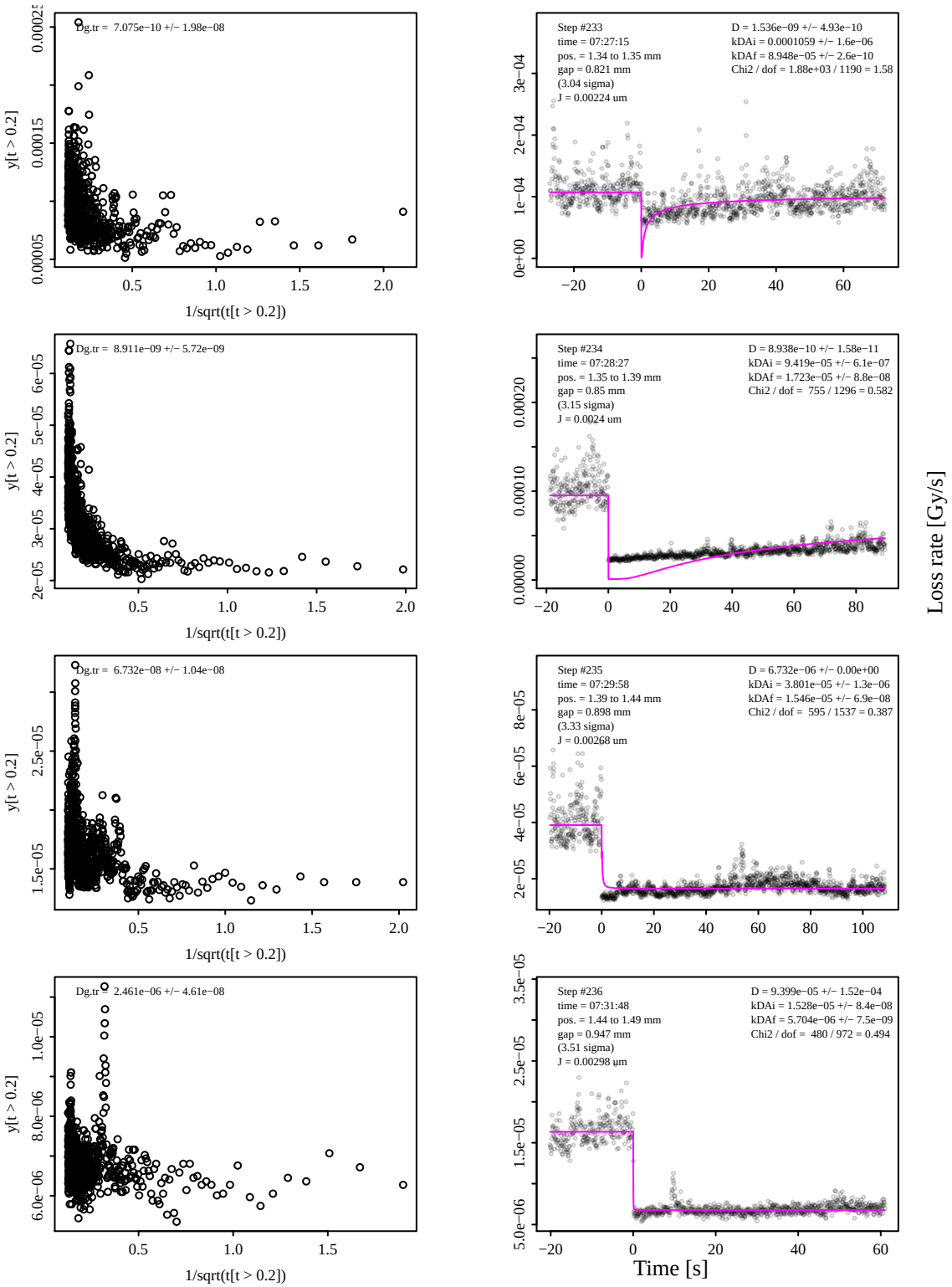

Dg.tr = 2.524e−10 +/− 2.99e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #213
time = 07:08:51
pos. = 1.58 to 1.56 mm
gap = 1.05 mm
(3.89 sigma)
J = 0.00365 um
D = 2.065e−10 +/− 4.79e−11
kDAi = 1.623e−05 +/− 6.3e−07
kDAf = 3.025e−05 +/− 0
Chi2 / dof = 2.84e+04 / 662 = 42.9
Dg.tr = 4.723e−10 +/− 5.22e−09
Step #214
time = 07:09:31
pos. = 1.56 to 1.54 mm
gap = 1.03 mm
(3.81 sigma)
J = 0.00352 um
D = 5.298e−11 +/− 6.63e−11
kDAi = 1.205e−05 +/− 4.2e−06
kDAf = 3.889e−05 +/− 0
Chi2 / dof = 2.53e+05 / 539 = 469
Dg.tr = 4.772e−10 +/− 8.95e−09
Step #215
time = 07:10:11
pos. = 1.54 to 1.52 mm
gap = 1.01 mm
(3.74 sigma)
J = 0.00338 um
D = 2.754e−10 +/− 7.16e−11
kDAi = 2.615e−05 +/− 1.3e−06
kDAf = 4.412e−05 +/− 0
Chi2 / dof = 1.92e+05 / 645 = 298
Dg.tr = 2.867e−10 +/− 6.10e−09
Step #216
time = 07:10:51
pos. = 1.52 to 1.5 mm
gap = 0.989 mm
(3.67 sigma)
J = 0.00325 um
D = 1.188e−10 +/− 6.19e−11
kDAi = 2.851e−05 +/− 2.2e−06
kDAf = 6.178e−05 +/− 0
Chi2 / dof = 4.63e+04 / 607 = 76.4
Loss rate [Gy/s]
Time [s]

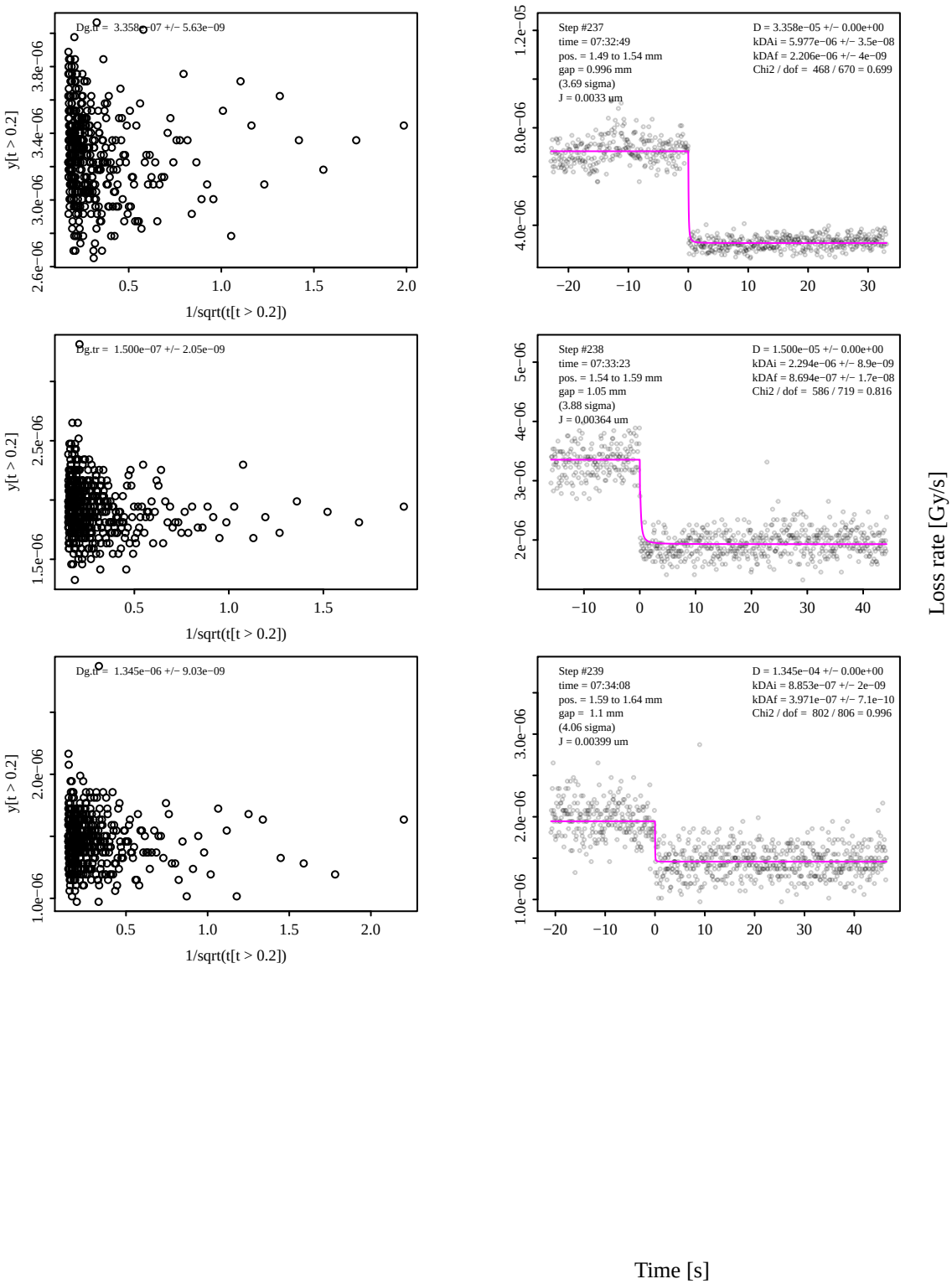

Dg.tr = 4.950e−10 +/− 1.06e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #217
time = 07:11:31
pos. = 1.5 to 1.48 mm
gap = 0.969 mm
(3.59 sigma)
J = 0.00312 um
D = 1.709e−10 +/− 3.10e−11
kDAi = 3.517e−05 +/− 1.5e−06
kDAf = 6.993e−05 +/− 0
Chi2 / dof = 2.48e+04 / 618 = 40.1
Dg.tr = 2.839e−10 +/− 7.53e−09
Step #218
time = 07:12:11
pos. = 1.48 to 1.46 mm
gap = 0.949 mm
(3.52 sigma)
J = 0.00299 um
D = 2.248e−10 +/− 2.91e−11
kDAi = 4.787e−05 +/− 9.5e−07
kDAf = 9.028e−05 +/− 0
Chi2 / dof = 4.42e+04 / 662 = 66.8
Dg.tr = 4.012e−10 +/− 1.68e−08
Step #219
time = 07:12:51
pos. = 1.46 to 1.44 mm
gap = 0.929 mm
(3.44 sigma)
J = 0.00287 um
D = 1.790e−10 +/− 7.12e−11
kDAi = 6.237e−05 +/− 5.4e−06
kDAf = 0.0001186 +/− 0
Chi2 / dof = 1.06e+05 / 627 = 170
Dg.tr = 3.173e−10 +/− 1.12e−08
Step #220
time = 07:13:31
pos. = 1.44 to 1.42 mm
gap = 0.91 mm
(3.37 sigma)
J = 0.00275 um
D = 1.644e−10 +/− 3.28e−11
kDAi = 6.498e−05 +/− 2.6e−06
kDAf = 0.0001287 +/− 0
Chi2 / dof = 1.2e+04 / 612 = 19.6
Loss rate [Gy/s]
Time [s]


Dg.tr = 3.224e−10 +/− 1.19e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #221
time = 07:14:11
pos. = 1.42 to 1.4 mm
gap = 0.889 mm
(3.3 sigma)
J = 0.00263 um
D = 1.155e−10 +/− 2.45e−11
kDAi = 7.06e−05 +/− 3.3e−06
kDAf = 0.0001334 +/− 7.6e−18
Chi2 / dof = 3.67e+04 / 1155 = 31.8
Dg.tr = 4.291e−10 +/− 3.18e−08
Step #222
time = 07:15:34
pos. = 1.4 to 1.39 mm
gap = 0.875 mm
(3.24 sigma)
J = 0.00254 um
D = 5.715e−10 +/− 1.42e−10
kDAi = 0.000108 +/− 3e−06
kDAf = 0.0001392 +/− 0
Chi2 / dof = 2.73e+03 / 758 = 3.6
Dg.tr = 9.509e−11 +/− 2.23e−08
Step #223
time = 07:16:14
pos. = 1.39 to 1.38 mm
gap = 0.864 mm
(3.2 sigma)
J = 0.00248 um
D = 4.570e−10 +/− 1.36e−10
kDAi = 0.0001245 +/− 3.7e−06
kDAf = 0.000156 +/− 0
Chi2 / dof = 5.93e+03 / 627 = 9.45
Dg.tr = 2.455e−10 +/− 2.53e−08
Step #224
time = 07:16:54
pos. = 1.38 to 1.37 mm
gap = 0.855 mm
(3.17 sigma)
J = 0.00243 um
D = 3.888e−10 +/− 8.84e−11
kDAi = 0.000127 +/− 3.5e−06
kDAf = 0.0001639 +/− 0
Chi2 / dof = 5.56e+03 / 673 = 8.26
Loss rate [Gy/s]
Time [s]


Dg.tr = 9.608e−11 +/− 1.28e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #225
time = 07:17:34
pos. = 1.37 to 1.36 mm
gap = 0.844 mm
(3.13 sigma)
J = 0.00237 um
D = 1.510e−10 +/− 3.43e−11
kDAi = 0.0001162 +/− 2.6e−06
kDAf = 0.0001565 +/− 0
Chi2 / dof = 1.2e+04 / 1193 = 10.1
Dg.tr = 8.710e−10 +/− 1.30e−07
Step #226
time = 07:20:35
pos. = 1.36 to 1.35 mm
gap = 0.834 mm
(3.09 sigma)
J = 0.00231 um
D = 1.965e−10 +/− 3.28e−11
kDAi = 0.0001133 +/− 3e−06
kDAf = 0.0001834 +/− 0
Chi2 / dof = 5.72e+03 / 597 = 9.57
Dg.tr = 1.621e−10 +/− 3.95e−08
Step #227
time = 07:21:27
pos. = 1.34 to 1.33 mm
gap = 0.814 mm
(3.02 sigma)
J = 0.0022 um
D = 4.627e−10 +/− 1.77e−10
kDAi = 0.0001888 +/− 8.5e−06
kDAf = 0.0002168 +/− 0
Chi2 / dof = 2.89e+03 / 572 = 5.05
Dg.tr = 2.058e−11 +/− 1.26e−08
Step #228
time = 07:21:55
pos. = 1.33 to 1.32 mm
gap = 0.804 mm
(2.98 sigma)
J = 0.00215 um
D = 1.162e−10 +/− 2.80e−11
kDAi = 0.0001645 +/− 5e−06
kDAf = 0.0002393 +/− 0
Chi2 / dof = 3.66e+03 / 593 = 6.17
Loss rate [Gy/s]
Time [s]


Dg.tr = 6.494e−11 +/− 1.35e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #229
time = 07:22:32
pos. = 1.32 to 1.31 mm
gap = 0.794 mm
(2.94 sigma)
J = 0.0021 um
D = 6.556e−11 +/− 1.85e−11
kDAi = 0.0001427 +/− 8.2e−06
kDAf = 0.0002164 +/− 0
Chi2 / dof = 1.4e+04 / 1173 = 12
Dg.tr = 4.527e−10 +/− 2.81e−08
Step #230
time = 07:23:57
pos. = 1.31 to 1.32 mm
gap = 0.794 mm
(2.94 sigma)
J = 0.0021 um
D = 8.828e−10 +/− 1.52e−10
kDAi = 0.0001747 +/− 1.9e−06
kDAf = 0.0001325 +/− 1.2e−14
Chi2 / dof = 1.26e+03 / 1008 = 1.25
Dg.tr = 3.995e−10 +/− 2.85e−08
Step #231
time = 07:24:56
pos. = 1.32 to 1.33 mm
gap = 0.803 mm
(2.98 sigma)
J = 0.00215 um
D = 1.347e−09 +/− 2.04e−10
kDAi = 0.0001418 +/− 1.8e−06
kDAf = 0.000116 +/− 9.2e−14
Chi2 / dof = 1.02e+03 / 771 = 1.32
Dg.tr = 6.799e−10 +/− 1.96e−08
Step #232
time = 07:25:39
pos. = 1.33 to 1.34 mm
gap = 0.813 mm
(3.01 sigma)
J = 0.0022 um
D = 9.509e−10 +/− 1.12e−10
kDAi = 0.0001196 +/− 1.4e−06
kDAf = 9.949e−05 +/− 8.9e−11
Chi2 / dof = 994 / 1281 = 0.776
Loss rate [Gy/s]
Time [s]


Dg.tr = 7.075e−10 +/− 1.98e−08
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #233
time = 07:27:15
pos. = 1.34 to 1.35 mm
gap = 0.821 mm
(3.04 sigma)
J = 0.00224 um
D = 1.536e−09 +/− 4.93e−10
kDAi = 0.0001059 +/− 1.6e−06
kDAf = 8.948e−05 +/− 2.6e−10
Chi2 / dof = 1.88e+03 / 1190 = 1.58
Dg.tr = 8.911e−09 +/− 5.72e−09
Step #234
time = 07:28:27
pos. = 1.35 to 1.39 mm
gap = 0.85 mm
(3.15 sigma)
J = 0.0024 um
D = 8.938e−10 +/− 1.58e−11
kDAi = 9.419e−05 +/− 6.1e−07
kDAf = 1.723e−05 +/− 8.8e−08
Chi2 / dof = 755 / 1296 = 0.582
Dg.tr = 6.732e−08 +/− 1.04e−08
Step #235
time = 07:29:58
pos. = 1.39 to 1.44 mm
gap = 0.898 mm
(3.33 sigma)
J = 0.00268 um
D = 6.732e−06 +/− 0.00e+00
kDAi = 3.801e−05 +/− 1.3e−06
kDAf = 1.546e−05 +/− 6.9e−08
Chi2 / dof = 595 / 1537 = 0.387
Dg.tr = 2.461e−06 +/− 4.61e−08
Step #236
time = 07:31:48
pos. = 1.44 to 1.49 mm
gap = 0.947 mm
(3.51 sigma)
J = 0.00298 um
D = 9.399e−05 +/− 1.52e−04
kDAi = 1.528e−05 +/− 8.4e−08
kDAf = 5.704e−06 +/− 7.5e−09
Chi2 / dof = 480 / 972 = 0.494
Loss rate [Gy/s]
Time [s]


Dg.tr = 3.358e−07 +/− 5.63e−09
y[t > 0.2]
1/sqrt(t[t > 0.2])
Step #237
time = 07:32:49
pos. = 1.49 to 1.54 mm
gap = 0.996 mm
(3.69 sigma)
J = 0.0033 um
D = 3.358e−05 +/− 0.00e+00
kDAi = 5.977e−06 +/− 3.5e−08
kDAf = 2.206e−06 +/− 4e−09
Chi2 / dof = 468 / 670 = 0.699
Dg.tr = 1.500e−07 +/− 2.05e−09
Step #238
time = 07:33:23
pos. = 1.54 to 1.59 mm
gap = 1.05 mm
(3.88 sigma)
J = 0.00364 um
D = 1.500e−05 +/− 0.00e+00
kDAi = 2.294e−06 +/− 8.9e−09
kDAf = 8.694e−07 +/− 1.7e−08
Chi2 / dof = 586 / 719 = 0.816
Dg.tr = 1.345e−06 +/− 9.03e−09
Step #239
time = 07:34:08
pos. = 1.59 to 1.64 mm
gap = 1.1 mm
(4.06 sigma)
J = 0.00399 um
D = 1.345e−04 +/− 0.00e+00
kDAi = 8.853e−07 +/− 2e−09
kDAf = 3.971e−07 +/− 7.1e−10
Chi2 / dof = 802 / 806 = 0.996
Loss rate [Gy/s]
Time [s]

Appendix B: Table of collimator steps

	Unix time [s]	Local time	Size [μm]	Init. half gap [mm]	Collisions?	Beam/plane
1	1340337548.833	05:59:08	−20.9	0.992	F	B1V
2	1340337575.097	05:59:35	−9.81	0.971	F	B1V
3	1340337622.708	06:00:22	−9.71	0.962	F	B1V
4	1340337632.739	06:00:32	−10.1	0.952	F	B1V
5	1340337642.739	06:00:42	−10.1	0.942	F	B1V
6	1340337652.752	06:00:52	−10.2	0.932	F	B1V
7	1340337662.691	06:01:02	−9.91	0.922	F	B1V
8	1340337686.789	06:01:26	−10.5	0.912	F	B1V
9	1340337701.848	06:01:41	−10.3	0.901	F	B1V
10	1340337716.774	06:01:56	−10.3	0.89	F	B1V
11	1340337731.791	06:02:11	−9.43	0.88	F	B1V
12	1340337746.774	06:02:26	−10.1	0.87	F	B1V
13	1340337761.786	06:02:41	−10.2	0.861	F	B1V
14	1340337776.790	06:02:56	−9.51	0.85	F	B1V
15	1340337791.754	06:03:11	−9.91	0.841	F	B1V
16	1340337806.767	06:03:26	−10.5	0.831	F	B1V
17	1340337821.790	06:03:41	−10.2	0.82	F	B1V
18	1340337844.426	06:04:04	−9.79	0.81	F	B1V
19	1340337859.398	06:04:19	−9.51	0.8	F	B1V
20	1340337874.402	06:04:34	−9.85	0.791	F	B1V
21	1340337889.443	06:04:49	−9.75	0.781	F	B1V
22	1340337904.430	06:05:04	−10.2	0.771	F	B1V
23	1340337919.432	06:05:19	−10.1	0.76	F	B1V
24	1340337934.432	06:05:34	−9.71	0.75	F	B1V
25	1340337949.433	06:05:49	−10.3	0.74	F	B1V
26	1340337964.424	06:06:04	−9.73	0.73	F	B1V
27	1340337979.430	06:06:19	−8.95	0.72	F	B1V
28	1340338015.612	06:06:55	−9.95	0.711	F	B1V
29	1340338030.615	06:07:10	−10	0.701	F	B1V
30	1340338098.519	06:08:18	−9.47	0.691	F	B1V
31	1340338123.509	06:08:43	−9.34	0.681	F	B1V
32	1340338148.513	06:09:08	−10	0.671	F	B1V
33	1340338173.509	06:09:33	−10	0.661	F	B1V
34	1340338198.526	06:09:58	−9.97	0.651	F	B1V

35	1340338275.519	06:11:15	−4.63	0.641	F	B1V
36	1340338300.591	06:11:40	−5.51	0.636	F	B1V
37	1340338325.565	06:12:05	−3.89	0.631	F	B1V
38	1340338350.582	06:12:30	−5.31	0.627	F	B1V
39	1340338375.487	06:12:55	−4.8	0.621	F	B1V
40	1340338400.532	06:13:20	−5.56	0.617	F	B1V
41	1340338425.599	06:13:45	−4.55	0.611	F	B1V
42	1340338450.591	06:14:10	−5.27	0.607	F	B1V
43	1340338500.534	06:15:00	−5.46	0.597	F	B1V
44	1340338638.009	06:17:18	−4.42	0.592	F	B1V
45	1340338678.012	06:17:58	−6.08	0.587	F	B1V
46	1340338718.016	06:18:38	−4.59	0.581	F	B1V
47	1340338758.016	06:19:18	−5.38	0.577	F	B1V
48	1340338798.015	06:19:58	−4.6	0.571	F	B1V
49	1340338838.009	06:20:38	−5.4	0.566	F	B1V
50	1340338878.018	06:21:18	−4.73	0.561	F	B1V
51	1340338918.033	06:21:58	−5.12	0.556	F	B1V
52	1340338958.022	06:22:38	−4.15	0.551	F	B1V
53	1340338998.024	06:23:18	−5.12	0.546	F	B1V
54	1340339059.019	06:24:19	−4.71	0.541	F	B1V
55	1340339099.046	06:24:59	−5.65	0.537	F	B1V
56	1340339139.035	06:25:39	−4.67	0.531	F	B1V
57	1340339179.020	06:26:19	−5.49	0.527	F	B1V
58	1340339219.041	06:26:59	−4.72	0.521	F	B1V
59	1340339259.059	06:27:39	−5.16	0.517	F	B1V
60	1340339339.012	06:28:59	−5.25	0.507	F	B1V
61	1340339379.013	06:29:39	−4.53	0.502	F	B1V
62	1340339419.036	06:30:19	−5.53	0.497	F	B1V
63	1340339590.660	06:33:10	17.6	0.491	F	B1V
64	1340339630.617	06:33:50	18.4	0.509	F	B1V
65	1340339873.123	06:37:53	17.7	0.527	F	B1V
66	1340339934.797	06:38:54	18.9	0.544	F	B1V
67	1340339965.834	06:39:25	94.2	0.564	F	B1V
68	1340340025.845	06:40:25	98	0.658	F	B1V
69	1340341295.191	07:01:35	−19.8	1.13	T	B1V
70	1340341310.200	07:01:50	−19.7	1.11	T	B1V
71	1340341325.193	07:02:05	−19.8	1.09	T	B1V

72	1340341340.197	07:02:20	−21.2	1.07	T	B1V
73	1340341355.185	07:02:35	−20.4	1.05	T	B1V
74	1340341370.193	07:02:50	−20.5	1.03	T	B1V
75	1340341385.197	07:03:05	−20	1.01	T	B1V
76	1340341400.202	07:03:20	−20.5	0.991	T	B1V
77	1340341415.185	07:03:35	−20.4	0.97	T	B1V
78	1340341430.189	07:03:50	−19.8	0.949	T	B1V
79	1340341447.935	07:04:07	−20	0.93	T	B1V
80	1340341462.935	07:04:22	−20.6	0.91	T	B1V
81	1340341477.957	07:04:37	−19.9	0.889	T	B1V
82	1340341506.378	07:05:06	−20.4	0.869	T	B1V
83	1340341531.437	07:05:31	−19.9	0.849	T	B1V
84	1340341556.448	07:05:56	−20.3	0.829	T	B1V
85	1340341581.392	07:06:21	−20.2	0.809	T	B1V
86	1340341606.367	07:06:46	−19.6	0.789	T	B1V
87	1340341631.366	07:07:11	−19.9	0.769	T	B1V
88	1340341656.359	07:07:36	−20	0.749	T	B1V
89	1340341681.364	07:08:01	−19.5	0.729	T	B1V
90	1340341706.371	07:08:26	−19.8	0.71	T	B1V
91	1340341731.377	07:08:51	−19.7	0.69	T	B1V
92	1340341775.047	07:09:35	−19.9	0.67	T	B1V
93	1340341800.040	07:10:00	−19.9	0.65	T	B1V
94	1340341847.691	07:10:47	−20	0.63	T	B1V
95	1340341887.698	07:11:27	−19.5	0.61	T	B1V
96	1340341927.705	07:12:07	−20.5	0.591	T	B1V
97	1340341991.739	07:13:11	−9.6	0.57	T	B1V
98	1340342031.713	07:13:51	−9.77	0.561	T	B1V
99	1340342071.691	07:14:31	−9.68	0.551	T	B1V
100	1340342111.693	07:15:11	−20.4	0.541	T	B1V
101	1340342151.675	07:15:51	−14.2	0.521	T	B1V
102	1340342191.702	07:16:31	−10.4	0.507	T	B1V
103	1340342231.681	07:17:11	−9.98	0.496	T	B1V
104	1340342271.696	07:17:51	−10.6	0.486	T	B1V
105	1340342383.511	07:19:43	−5.64	0.476	T	B1V
106	1340342413.192	07:20:13	−4.51	0.47	T	B1V
107	1340342441.185	07:20:41	−5.71	0.466	T	B1V
108	1340342475.084	07:21:15	−5.13	0.46	T	B1V

109	1340342493.590	07:21:33	−5.17	0.454	T	B1V
110	1340342527.377	07:22:07	−4.42	0.449	T	B1V
111	1340342547.110	07:22:27	−5.29	0.445	T	B1V
112	1340342606.709	07:23:26	8.72	0.44	T	B1V
113	1340342678.828	07:24:38	9.35	0.448	T	B1V
114	1340342774.161	07:26:14	9.29	0.457	T	B1V
115	1340342857.830	07:27:37	9.54	0.467	T	B1V
116	1340342923.165	07:28:43	9.22	0.476	T	B1V
117	1340342973.231	07:29:33	18.5	0.486	T	B1V
118	1340343099.747	07:31:39	19.1	0.504	T	B1V
119	1340343159.647	07:32:39	18.3	0.523	T	B1V
120	1340343209.508	07:33:29	18.6	0.542	T	B1V
121	1340343243.045	07:34:03	47.7	0.56	T	B1V
122	1340338087.014	06:08:07	−10	1.54	F	B2H
123	1340338097.009	06:08:17	−9.04	1.53	F	B2H
124	1340338106.932	06:08:26	−10.9	1.52	F	B2H
125	1340338116.975	06:08:36	−10.5	1.51	F	B2H
126	1340338126.971	06:08:46	−10.7	1.5	F	B2H
127	1340338137.011	06:08:57	−8.15	1.49	F	B2H
128	1340338146.999	06:09:06	−10.4	1.48	F	B2H
129	1340338156.968	06:09:16	−9.9	1.47	F	B2H
130	1340338210.518	06:10:10	−10.6	1.46	F	B2H
131	1340338220.314	06:10:20	−9.06	1.45	F	B2H
132	1340338230.574	06:10:30	−10.5	1.44	F	B2H
133	1340338240.582	06:10:40	−9.85	1.43	F	B2H
134	1340338250.537	06:10:50	−10.9	1.42	F	B2H
135	1340338260.531	06:11:00	−8.79	1.41	F	B2H
136	1340338270.550	06:11:10	−11	1.4	F	B2H
137	1340338280.530	06:11:20	−9.58	1.39	F	B2H
138	1340338290.549	06:11:30	−10.4	1.38	F	B2H
139	1340338300.589	06:11:40	−9.29	1.37	F	B2H
140	1340338310.553	06:11:50	−10.9	1.36	F	B2H
141	1340338320.522	06:12:00	−9.87	1.35	F	B2H
142	1340338330.557	06:12:10	−10.8	1.34	F	B2H
143	1340338340.596	06:12:20	−8.92	1.33	F	B2H
144	1340338350.555	06:12:30	−10.4	1.32	F	B2H
145	1340338360.427	06:12:40	−9.61	1.31	F	B2H

146	1340338370.427	06:12:50	−11	1.3	F	B2H
147	1340338381.167	06:13:01	−9.39	1.29	F	B2H
148	1340338396.177	06:13:16	−10.5	1.28	F	B2H
149	1340338411.190	06:13:31	−9.77	1.27	F	B2H
150	1340338426.192	06:13:46	−10.5	1.26	F	B2H
151	1340338441.199	06:14:01	−9.19	1.25	F	B2H
152	1340338456.202	06:14:16	−10.5	1.24	F	B2H
153	1340338471.190	06:14:31	−9.75	1.23	F	B2H
154	1340338486.198	06:14:46	−10	1.22	F	B2H
155	1340338501.197	06:15:01	−9.17	1.21	F	B2H
156	1340338516.194	06:15:16	−10.5	1.2	F	B2H
157	1340338531.182	06:15:31	−9.94	1.19	F	B2H
158	1340338546.209	06:15:46	−10.6	1.18	F	B2H
159	1340338561.127	06:16:01	−9.08	1.17	F	B2H
160	1340338631.097	06:17:11	−11.1	1.16	F	B2H
161	1340338646.099	06:17:26	−9.9	1.15	F	B2H
162	1340338661.004	06:17:41	−10.8	1.14	F	B2H
163	1340338676.034	06:17:56	−8.83	1.13	F	B2H
164	1340338690.695	06:18:10	−10.8	1.12	F	B2H
165	1340338705.418	06:18:25	−9.08	1.11	F	B2H
166	1340338721.008	06:18:41	−10.5	1.1	F	B2H
167	1340338736.109	06:18:56	−9.4	1.09	F	B2H
168	1340338751.042	06:19:11	−10.8	1.08	F	B2H
169	1340338766.021	06:19:26	−9.8	1.07	F	B2H
170	1340338781.010	06:19:41	−10.9	1.06	F	B2H
171	1340338934.452	06:22:14	−10.7	1.03	F	B2H
172	1340338954.396	06:22:34	−10.4	1.02	F	B2H
173	1340338974.380	06:22:54	−9.93	1.01	F	B2H
174	1340338994.445	06:23:14	−9.97	1	F	B2H
175	1340339014.396	06:23:34	−10.2	0.994	F	B2H
176	1340339034.423	06:23:54	−10.3	0.983	F	B2H
177	1340339054.368	06:24:14	−9.82	0.973	F	B2H
178	1340339074.351	06:24:34	−9.97	0.963	F	B2H
179	1340339094.408	06:24:54	−10.3	0.953	F	B2H
180	1340339141.531	06:25:41	−9.5	0.944	F	B2H
181	1340339181.511	06:26:21	−10.5	0.934	F	B2H
182	1340339221.530	06:27:01	−9.39	0.923	F	B2H

183	1340339261.532	06:27:41	−10.3	0.914	F	B2H
184	1340339301.522	06:28:21	−9.52	0.904	F	B2H
185	1340339341.570	06:29:01	−9.92	0.894	F	B2H
186	1340339381.523	06:29:41	−10.3	0.884	F	B2H
187	1340339421.505	06:30:21	−10.4	0.874	F	B2H
188	1340339461.493	06:31:01	−9.78	0.864	F	B2H
189	1340339501.514	06:31:41	−10.3	0.854	F	B2H
190	1340339614.119	06:33:34	−10.3	0.844	F	B2H
191	1340339654.108	06:34:14	−10.1	0.834	F	B2H
192	1340339804.898	06:36:44	17.9	0.824	F	B2H
193	1340339889.045	06:38:09	19	0.842	F	B2H
194	1340339980.444	06:39:40	18.3	0.861	F	B2H
195	1340340141.665	06:42:21	96.4	0.88	F	B2H
196	1340340179.750	06:42:59	99.6	0.976	F	B2H
197	1340340191.594	06:43:11	99.5	1.08	F	B2H
198	1340341318.670	07:01:58	−20.5	1.36	T	B2H
199	1340341333.670	07:02:13	−19.1	1.34	T	B2H
200	1340341348.676	07:02:28	−20.4	1.32	T	B2H
201	1340341363.678	07:02:43	−19.9	1.3	T	B2H
202	1340341378.653	07:02:58	−20	1.28	T	B2H
203	1340341393.626	07:03:13	−20.1	1.26	T	B2H
204	1340341408.668	07:03:28	−20.4	1.24	T	B2H
205	1340341423.663	07:03:43	−18.7	1.22	T	B2H
206	1340341438.661	07:03:58	−20.9	1.2	T	B2H
207	1340341453.620	07:04:13	−19.7	1.18	T	B2H
208	1340341523.097	07:05:23	−20.3	1.16	T	B2H
209	1340341548.108	07:05:48	−19.6	1.14	T	B2H
210	1340341573.069	07:06:13	−20.5	1.12	T	B2H
211	1340341598.112	07:06:38	−19.6	1.1	T	B2H
212	1340341691.297	07:08:11	−20.8	1.08	T	B2H
213	1340341731.450	07:08:51	−19.6	1.06	T	B2H
214	1340341771.501	07:09:31	−20.8	1.04	T	B2H
215	1340341811.507	07:10:11	−19.5	1.02	T	B2H
216	1340341851.453	07:10:51	−20.1	0.999	T	B2H
217	1340341891.493	07:11:31	−20.1	0.979	T	B2H
218	1340341931.408	07:12:11	−20.6	0.959	T	B2H
219	1340341971.479	07:12:51	−19.3	0.939	T	B2H

220	1340342011.508	07:13:31	−20.5	0.92	T	B2H
221	1340342051.508	07:14:11	−19.4	0.899	T	B2H
222	1340342134.323	07:15:34	−10.7	0.88	T	B2H
223	1340342174.242	07:16:14	−9.67	0.869	T	B2H
224	1340342214.375	07:16:54	−10.7	0.86	T	B2H
225	1340342254.062	07:17:34	−9.47	0.849	T	B2H
226	1340342435.448	07:20:35	−10.9	0.839	T	B2H
227	1340342487.835	07:21:27	−10.2	0.819	T	B2H
228	1340342515.264	07:21:55	−9.04	0.809	T	B2H
229	1340342552.683	07:22:32	−10.5	0.799	T	B2H
230	1340342637.579	07:23:57	9.91	0.789	T	B2H
231	1340342696.352	07:24:56	9.03	0.799	T	B2H
232	1340342739.892	07:25:39	9.6	0.808	T	B2H
233	1340342835.020	07:27:15	9.2	0.816	T	B2H
234	1340342907.989	07:28:27	48.9	0.826	T	B2H
235	1340342998.485	07:29:58	47.4	0.874	T	B2H
236	1340343108.127	07:31:48	50	0.922	T	B2H
237	1340343169.989	07:32:49	48.7	0.971	T	B2H
238	1340343203.540	07:33:23	50.6	1.02	T	B2H
239	1340343248.766	07:34:08	48.8	1.07	T	B2H

TABLE IV: Table of collimator steps (negative size means inward).

Appendix C: Table of fit results for each step

	Local time	Action [μm]	Half gap [σ]	D [μm^2/s]	δD	kDA_i [Gy/s]	δkDA_i	kDA_f	δkDA_f	χ^2	dof
1	05:59:08	0.00615	6.29	1.13e-11	3e-12	6e-08	6e-09	5.7e-07	0	1.6e+05	534
2	05:59:35	0.00596	6.19	1.94e-11	1.2e-12	1.3e-07	0	4.4e-07	0	4.5e+04	641
3	06:00:22	0.00584	6.13	2.54e-11	1.2e-11	3.1e-07	2e-10	1.1e-06	0	9.6e+04	373
4	06:00:32	0.00572	6.07	1.25e-11	2.8e-12	2.9e-07	0	1.1e-06	0	1.3e+05	147
5	06:00:42	0.0056	6	1.29e-11	3.5e-12	3.2e-07	0	1.7e-06	0	1.7e+05	155
6	06:00:52	0.00548	5.94	3.35e-11	1.6e-11	8.2e-07	6e-08	2e-06	0	5e+04	161
7	06:01:02	0.00536	5.87	7.59e-11	1.6e-11	5.2e-07	0	1.3e-06	0	4.6e+05	313
8	06:01:26	0.00524	5.81	2.25e-11	3.3e-11	5.6e-07	1e-07	1.1e-06	0	8.1e+05	258
9	06:01:41	0.00512	5.74	3.6e-12	5.2e-13	2.5e-07	0	2.6e-06	0	3.8e+05	207
10	06:01:56	0.005	5.67	1.15e-10	9.1e-11	1.5e-06	2e-07	2.5e-06	0	3.6e+05	242
11	06:02:11	0.00489	5.61	4.66e-11	1.7e-10	1.5e-06	7e-07	4.2e-06	0	2e+05	228
12	06:02:26	0.00478	5.55	8.9e-11	3.7e-10	1.8e-06	1e-06	5e-06	0	3.9e+05	231
13	06:02:41	0.00467	5.48	1.86e-10	3.2e-10	2.7e-06	8e-07	5.4e-06	0	1.3e+05	243
14	06:02:56	0.00457	5.42	1.13e-11	2.8e-12	1.3e-06	0	5.9e-06	0	5.6e+05	212
15	06:03:11	0.00446	5.36	4.39e-11	5.2e-11	2.6e-06	6e-07	6.8e-06	0	7.6e+05	229
16	06:03:26	0.00435	5.29	4.37e-11	4.7e-11	1.8e-06	0	3.8e-06	0	2.5e+05	217
17	06:03:41	0.00424	5.22	3.03e-12	9.1e-13	9.7e-07	0	7.7e-06	0	5.1e+05	322
18	06:04:04	0.00414	5.16	2.59e-11	7e-12	4e-06	4e-07	1.4e-05	0	8.7e+05	245
19	06:04:19	0.00404	5.1	6.54e-11	1.7e-10	7.5e-06	3e-06	1.7e-05	0	5.9e+05	223
20	06:04:34	0.00394	5.03	2.72e-10	2.1e-09	1.1e-05	4e-06	1.7e-05	0	3.5e+05	229
21	06:04:49	0.00384	4.97	9.12e-11	3.6e-11	1.2e-05	2e-06	2.5e-05	0	5.6e+04	240
22	06:05:04	0.00374	4.91	9.07e-12	3.7e-12	4.2e-06	0	1.4e-05	0	8.7e+05	217
23	06:05:19	0.00364	4.84	1.06e-12	1.2e-13	2.3e-06	0	3.6e-05	0	6.4e+05	204
24	06:05:34	0.00355	4.78	3.39e-12	9.4e-13	5e-06	0	3.6e-05	0	4.4e+04	199
25	06:05:49	0.00345	4.71	1.78e-11	4.4e-12	9.6e-06	0	3.2e-05	0	4e+05	231
26	06:06:04	0.00336	4.65	5.1e-12	5e-13	8.1e-06	0	5.4e-05	0	6.7e+05	221
27	06:06:19	0.00327	4.59	8.31e-11	2e-11	1.7e-05	0	2.1e-05	0	2.7e+05	487
28	06:06:55	0.00318	4.52	2.97e-11	5.9e-12	1.4e-05	4e-07	4.4e-05	0	7.6e+05	291
29	06:07:10	0.00309	4.46	1.44e-11	1.2e-12	9.7e-06	0	2.2e-05	2e-17	4.7e+04	864
30	06:08:18	0.00301	4.4	1.1e-11	2e-12	1.3e-05	5e-07	5.3e-05	0	9.3e+04	491
31	06:08:43	0.00292	4.34	1.07e-11	2.8e-14	1.4e-05	0	3.4e-05	0	1.5e+05	384
32	06:09:08	0.00284	4.27	2.39e-12	6.7e-13	8e-06	0	6.8e-05	0	2.6e+05	376
33	06:09:33	0.00275	4.21	6.12e-12	1.3e-12	1.4e-05	0	6.8e-05	0	3.9e+04	376
34	06:09:58	0.00267	4.14	1.07e-11	9.9e-13	1.5e-05	0	4.3e-05	7e-12	3.2e+05	794

35	06:11:15	0.00261	4.09	3.48e-11	5.9e-12	2.6e-05	5e-07	4.5e-05	0	5.1e+03	664
36	06:11:40	0.00256	4.06	3.03e-12	3.8e-12	1.7e-05	8e-06	6.8e-05	0	1.3e+04	362
37	06:12:05	0.00253	4.03	1.27e-10	1.9e-10	5.3e-05	9e-06	6.4e-05	0	2.9e+04	409
38	06:12:30	0.00249	4	2.06e-12	9.3e-13	1.6e-05	3e-06	7.8e-05	0	3e+04	373
39	06:12:55	0.00245	3.97	3.16e-11	3.9e-12	4.6e-05	8e-07	7e-05	0	8.4e+03	373
40	06:13:20	0.00241	3.93	1.23e-11	5.5e-12	3.9e-05	4e-06	0.0001	0	1.2e+05	384
41	06:13:45	0.00237	3.9	7.46e-11	1.3e-11	6.2e-05	1e-06	9e-05	0	4.7e+03	384
42	06:14:10	0.00233	3.87	3.03e-12	8.8e-13	2.6e-05	7e-08	9.4e-05	0	1.5e+04	389
43	06:15:00	0.00225	3.81	1.94e-12	9.6e-14	2.4e-05	0	8.9e-05	0	5.9e+04	599
44	06:17:18	0.00222	3.78	3.01e-11	3.3e-12	4.1e-05	1e-06	7.9e-05	0	1.3e+04	569
45	06:17:58	0.00218	3.74	1.16e-10	6.9e-11	5e-05	4e-06	6.4e-05	0	1.3e+04	637
46	06:18:38	0.00214	3.71	1.14e-11	1.7e-12	3.6e-05	2e-06	7.5e-05	0	1e+04	594
47	06:19:18	0.0021	3.68	1.26e-11	1.3e-11	4.6e-05	5e-06	8.2e-05	0	6.7e+04	559
48	06:19:58	0.00207	3.65	1.12e-10	7e-11	5.9e-05	3e-06	6.9e-05	4e-15	5.3e+03	656
49	06:20:38	0.00203	3.61	7.05e-12	1.7e-12	5e-05	3e-06	0.00012	0	5.1e+04	678
50	06:21:18	0.00199	3.58	2.23e-12	2e-13	1.7e-05	0	5.6e-05	0	1.4e+04	542
51	06:21:58	0.00196	3.55	3.2e-12	6.4e-13	3.9e-05	3e-06	0.00012	0	7.4e+04	640
52	06:22:38	0.00193	3.52	1.53e-11	7.1e-12	5.3e-05	8e-06	8.2e-05	0	1.1e+04	607
53	06:23:18	0.00189	3.48	4.5e-12	1.1e-12	3.6e-05	2e-06	0.00011	0	3.4e+04	499
54	06:24:19	0.00186	3.45	2.45e-11	5.8e-12	5.6e-05	2e-06	0.00011	0	1.5e+04	561
55	06:24:59	0.00182	3.42	1.82e-11	1.3e-11	5.6e-05	8e-06	8.9e-05	0	3.4e+04	664
56	06:25:39	0.00178	3.39	7.88e-11	2.9e-11	6.6e-05	3e-06	8.8e-05	0	3.5e+03	670
57	06:26:19	0.00175	3.36	3.2e-12	6.7e-13	5e-05	4e-06	0.00016	0	1.2e+04	591
58	06:26:59	0.00172	3.32	1.1e-10	2.3e-11	8.7e-05	3e-06	0.00012	0	1.8e+03	591
59	06:27:39	0.00169	3.29	1.41e-11	1.5e-11	7.6e-05	1e-05	0.00012	0	7.9e+04	668
60	06:28:59	0.00163	3.23	1.05e-11	8.4e-12	6.9e-05	1e-05	0.00012	0	2.6e+04	676
61	06:29:39	0.00159	3.2	9.03e-12	1.1e-12	8.3e-05	2e-06	0.00016	0	1.3e+04	662
62	06:30:19	0.00156	3.17	3.01e-12	6.3e-13	4.2e-05	0	7.1e-05	0	1.1e+05	1019
63	06:33:10	0.0016	3.2	6.3e-12	1.4e-16	5.2e-05	7e-07	4.1e-06	1e-18	2.4e+02	605
64	06:33:50	0.00171	3.32	1.14e-09	1.5e-10	2.2e-06	6e-08	4.6e-07	3e-10	3.1e+03	1523
65	06:37:53	0.00183	3.43	1.46e-06	0	2.1e-06	4e-09	3.8e-07	2e-08	6.2e+02	1226
66	06:38:54	0.00196	3.55	4e-06	5.8e-06	4.4e-07	6e-10	2.2e-07	3e-10	6.9e+02	809
67	06:39:25	0.0024	3.91	5.93e-06	0	2.5e-07	3e-08	1.8e-07	1e-10	5.8e+02	690
68	06:40:25	0.0032	4.53	1.89e-06	7.2e-07	1.8e-07	1e-10	1.8e-07	3e-10	7.5e+02	768
69	07:01:35	0.00804	7.19	4.37e-09	1.7e-09	4e-07	4e-09	6.6e-07	0	2.2e+03	335
70	07:01:50	0.00776	7.06	2.94e-09	2e-09	4.4e-07	6e-08	7.5e-07	0	1.1e+03	234
71	07:02:05	0.00748	6.94	1.21e-09	1.3e-09	4.4e-07	1e-07	9.5e-07	0	2e+03	237

72	07:02:20	0.0072	6.8	3.46e-09	2.5e-10	6.6e-07	8e-09	1.2e-06	0	3.1e+03	237
73	07:02:35	0.00692	6.67	2.54e-09	1.4e-09	7.5e-07	7e-08	1.3e-06	0	7.4e+03	237
74	07:02:50	0.00665	6.54	1.42e-09	1.2e-10	9.1e-07	3e-08	2e-06	0	2e+04	239
75	07:03:05	0.00639	6.41	2.26e-09	1.4e-09	1.3e-06	2e-07	2.4e-06	0	1.1e+04	228
76	07:03:20	0.00614	6.28	4e-09	3.1e-09	1.5e-06	3e-07	2.3e-06	0	4.2e+03	230
77	07:03:35	0.00589	6.15	4.65e-09	2.4e-09	1.7e-06	1e-07	2.3e-06	0	4.3e+03	247
78	07:03:50	0.00564	6.02	3.29e-09	1.1e-09	1.7e-06	1e-07	2.7e-06	0	6.2e+03	278
79	07:04:07	0.0054	5.89	3.07e-09	1e-09	2.1e-06	1e-07	3.4e-06	0	1.3e+04	253
80	07:04:22	0.00517	5.76	4.4e-09	2.9e-09	2.5e-06	3e-07	3.7e-06	0	1.6e+04	234
81	07:04:37	0.00494	5.63	1.75e-09	1.5e-09	2.1e-06	4e-07	3.4e-06	8e-18	4e+04	403
82	07:05:06	0.00472	5.51	3.03e-09	2.9e-10	2.6e-06	6e-08	3.9e-06	0	3.4e+04	395
83	07:05:31	0.0045	5.38	6.87e-10	1.8e-10	2e-06	2e-07	4.1e-06	0	2.6e+04	362
84	07:05:56	0.00428	5.25	5.57e-10	4.6e-10	2e-06	4e-07	4.6e-06	0	4.6e+04	346
85	07:06:21	0.00408	5.12	1.19e-09	5.9e-10	3.2e-06	4e-07	5.5e-06	0	9.5e+04	373
86	07:06:46	0.00388	4.99	9.66e-10	4.9e-10	4e-06	4e-07	6.3e-06	0	4.9e+04	411
87	07:07:11	0.00368	4.87	1.47e-09	9.9e-10	4.2e-06	6e-07	6.5e-06	0	5.7e+04	404
88	07:07:36	0.00349	4.74	5.38e-10	2.2e-10	4.3e-06	4e-07	7.3e-06	0	6.7e+04	409
89	07:08:01	0.00331	4.61	2.29e-10	5e-10	3.6e-06	2e-06	8.7e-06	0	2.4e+05	392
90	07:08:26	0.00313	4.48	4.23e-10	4.8e-11	6.6e-06	2e-07	1.3e-05	0	7.7e+04	321
91	07:08:51	0.00295	4.36	1.65e-10	1.1e-11	4.2e-06	0	6.9e-06	4e-06	1.5e+04	648
92	07:09:35	0.00278	4.23	2.65e-10	3.8e-11	7.1e-06	2e-07	1.6e-05	0	6e+05	495
93	07:10:00	0.00262	4.1	2.58e-11	9.4e-12	3.8e-06	4e-07	1.7e-05	0	3.7e+05	659
94	07:10:47	0.00246	3.97	2.22e-10	4.1e-11	9e-06	4e-07	1.9e-05	0	2.5e+05	648
95	07:11:27	0.0023	3.85	6.91e-11	2.7e-11	1.1e-05	1e-06	2.8e-05	0	1.7e+06	635
96	07:12:07	0.00215	3.72	7.18e-12	1.7e-13	4.9e-06	0	3.1e-05	0	1.7e+06	886
97	07:13:11	0.00204	3.62	4.85e-11	1.6e-11	1.8e-05	1e-06	3.6e-05	0	1e+04	659
98	07:13:51	0.00197	3.56	4.68e-11	1.6e-11	2.5e-05	2e-06	3.9e-05	4e-11	1.5e+04	700
99	07:14:31	0.0019	3.5	2.69e-11	2.3e-11	1.9e-05	4e-06	4e-05	0	1.1e+05	629
100	07:15:11	0.0018	3.4	1.77e-11	7.9e-12	2.1e-05	3e-06	0.0001	0	1.3e+07	577
101	07:15:51	0.00169	3.29	7.89e-12	6.5e-13	2.3e-05	0	9.7e-05	0	1.4e+05	582
102	07:16:31	0.00161	3.21	4.21e-11	1.2e-11	6e-05	3e-06	0.00012	0	1.2e+05	638
103	07:17:11	0.00154	3.15	2.32e-12	2.5e-13	3e-05	0	0.00014	0	3.4e+04	585
104	07:17:51	0.00148	3.08	2.4e-11	2.8e-11	5.2e-05	1e-05	0.00013	9e-05	1.2e+05	1479
105	07:19:43	0.00143	3.03	8.08e-12	1.4e-12	7.7e-05	3e-06	0.00018	0	8.1e+03	666
106	07:20:13	0.0014	3	3.37e-11	5.1e-12	0.00011	4e-06	0.00018	0	1.6e+03	455
107	07:20:41	0.00137	2.97	2.18e-12	2e-13	5.1e-05	0	0.00017	2e-13	2.2e+04	518
108	07:21:15	0.00134	2.93	2.09e-11	4.2e-12	9.9e-05	3e-06	0.0002	0	8.1e+03	329

109	07:21:33	0.0013	2.9	2.06e-12	4.2e-12	7.3e-05	4e-05	0.00026	0	1.3e+04	454
110	07:22:07	0.00128	2.86	2.98e-11	1.7e-12	0.00021	3e-06	0.00039	0	9e+03	354
111	07:22:27	0.00125	2.83	1.08e-11	1.3e-12	0.00011	0	0.00018	8e-14	7.7e+04	789
112	07:23:26	0.00126	2.84	4.51e-11	1.8e-12	0.00016	2e-06	5.5e-05	9e-10	6.9e+02	1107
113	07:24:38	0.00131	2.9	6.48e-11	5.3e-12	5e-05	1e-06	2.2e-05	6e-12	5.9e+02	1332
114	07:26:14	0.00136	2.96	1.69e-10	1e-11	2.4e-05	4e-07	1.4e-05	0	7.1e+02	1185
115	07:27:37	0.00142	3.02	2.01e-10	2.3e-11	1.4e-05	4e-07	8.1e-06	5e-10	7.9e+02	992
116	07:28:43	0.00148	3.08	4.96e-10	3e-11	8e-06	1e-07	5.4e-06	2e-16	5.3e+02	823
117	07:29:33	0.00156	3.17	5.36e-10	2.8e-11	5.7e-06	5e-08	2.7e-06	5e-12	5.5e+02	817
118	07:31:39	0.00169	3.29	1.16e-09	4.4e-11	3.2e-06	8e-09	1.7e-06	1e-08	1.8e+03	1534
119	07:32:39	0.00181	3.41	1.41e-09	9.2e-11	2e-06	7e-09	1.2e-06	4e-09	7.5e+02	796
120	07:33:29	0.00194	3.53	3.88e-09	7.6e-10	1.3e-06	4e-09	9.3e-07	2e-09	5.3e+02	591
121	07:34:03	0.00218	3.74	2.94e-06	8.1e-07	9.7e-07	1e-09	5.7e-07	8e-09	6.4e+02	640
122	06:08:07	0.00783	5.69	7.72e-10	0	1.3e-06	5e-08	2.3e-06	2e-13	2.6e+03	155
123	06:08:17	0.00773	5.65	4.51e-09	7.8e-10	2.1e-06	6e-08	2.4e-06	0	3.8e+02	152
124	06:08:26	0.00763	5.62	2.8e-10	1.4e-10	1.5e-06	1e-07	2.5e-06	0	1.3e+04	152
125	06:08:36	0.00752	5.58	5.24e-10	1.8e-10	2.1e-06	1e-07	3.7e-06	0	4e+03	155
126	06:08:46	0.00742	5.54	4.3e-10	2.7e-10	2.7e-06	3e-07	4.7e-06	0	5.4e+03	157
127	06:08:57	0.00732	5.5	1.57e-10	1.1e-10	3.3e-06	3e-07	5.5e-06	0	9.4e+03	147
128	06:09:06	0.00723	5.47	1.76e-09	0	6.3e-06	4e-07	2.5e-05	1e-06	1.6e+04	160
129	06:09:16	0.00713	5.43	1.22e-10	3.1e-11	2.8e-06	0	4.5e-06	0	3.4e+03	676
130	06:10:10	0.00704	5.39	1.06e-10	2.9e-11	3.7e-06	4e-08	1.1e-05	0	6.2e+03	354
131	06:10:20	0.00694	5.36	3.43e-10	1.7e-10	7.5e-06	5e-07	1.3e-05	0	3.2e+04	157
132	06:10:30	0.00685	5.32	2.12e-11	3.2e-11	4.9e-06	3e-06	1.8e-05	0	5.1e+03	147
133	06:10:40	0.00675	5.28	1.78e-10	6.8e-11	9.4e-06	6e-07	2e-05	0	1.7e+04	147
134	06:10:50	0.00665	5.24	2.48e-10	7.4e-10	1.2e-05	4e-06	2e-05	0	1.6e+04	149
135	06:11:00	0.00656	5.21	9.61e-11	1.5e-11	1.2e-05	8e-07	3.1e-05	0	8e+03	152
136	06:11:10	0.00647	5.17	7.19e-11	1.9e-10	1.3e-05	6e-06	2.7e-05	0	7.1e+03	152
137	06:11:20	0.00637	5.13	3.07e-10	8.8e-11	2.1e-05	1e-06	3.8e-05	0	2.2e+04	158
138	06:11:30	0.00628	5.1	9.41e-10	3.7e-10	3e-05	1e-06	3.9e-05	0	5.6e+03	166
139	06:11:40	0.00619	5.06	3.29e-10	3.2e-10	2.2e-05	4e-06	3.4e-05	0	1.4e+03	161
140	06:11:50	0.0061	5.02	1.59e-10	2.3e-11	2.9e-05	1e-06	5.7e-05	0	7.6e+04	163
141	06:12:00	0.00601	4.98	2.25e-10	2.2e-10	3e-05	7e-06	5.4e-05	0	3e+03	152
142	06:12:10	0.00592	4.95	1.03e-09	1.5e-09	4.5e-05	4e-06	5.2e-05	0	9.5e+03	174
143	06:12:20	0.00583	4.91	2.3e-10	5.2e-11	4.1e-05	4e-06	6.3e-05	0	8.1e+03	172
144	06:12:30	0.00574	4.87	5.22e-10	4.8e-10	5.1e-05	3e-06	7.4e-05	0	1.4e+04	160
145	06:12:40	0.00566	4.84	3.06e-10	7.1e-11	5.2e-05	4e-06	7e-05	0	9.2e+03	155

146	06:12:50	0.00557	4.8	6e-05	0	5.9e-05	3e-06	0.00011	4e-06	1.9e+04	163
147	06:13:01	0.00548	4.76	1.96e-09	1.2e-09	6.8e-05	5e-06	6.6e-05	0	1.5e+03	240
148	06:13:16	0.0054	4.72	6.05e-10	3.6e-10	5.1e-05	3e-06	6.3e-05	0	4.5e+03	266
149	06:13:31	0.00531	4.69	2.46e-10	5.1e-11	5.1e-05	1e-06	6.9e-05	0	4.7e+03	231
150	06:13:46	0.00523	4.65	3.13e-10	4.4e-11	4.9e-05	3e-06	8.9e-05	0	6.2e+03	237
151	06:14:01	0.00515	4.61	2.74e-10	8.1e-11	5.8e-05	3e-06	9.1e-05	0	5.6e+03	245
152	06:14:16	0.00507	4.58	4.94e-10	2e-10	7.4e-05	3e-06	9.4e-05	0	3.6e+03	266
153	06:14:31	0.00498	4.54	4.06e-10	1.5e-10	5.5e-05	1e-06	8.1e-05	0	8e+02	239
154	06:14:46	0.0049	4.5	1.57e-10	3.6e-11	6.9e-05	1e-06	0.00011	0	2.5e+04	226
155	06:15:01	0.00482	4.46	4.39e-10	1.8e-10	6.9e-05	7e-06	8.1e-05	0	1.7e+03	229
156	06:15:16	0.00474	4.43	2.23e-10	7.5e-11	6.5e-05	5e-06	9.6e-05	0	1.5e+04	236
157	06:15:31	0.00466	4.39	3.04e-10	4e-11	7.8e-05	4e-06	0.00012	0	2.1e+03	263
158	06:15:46	0.00458	4.35	3.59e-10	2e-10	7.9e-05	8e-06	0.00012	0	2.9e+03	234
159	06:16:01	0.0045	4.31	4.3e-11	2.8e-12	3.6e-05	0	6.7e-05	3e-13	7.9e+03	894
160	06:17:11	0.00443	4.28	1.75e-10	8e-12	3.8e-05	3e-07	8.2e-05	0	1.8e+04	303
161	06:17:26	0.00435	4.24	2.51e-10	3.4e-11	7e-05	1e-06	0.0001	0	4.3e+03	256
162	06:17:41	0.00427	4.2	2.63e-10	4.8e-11	8.8e-05	2e-06	0.00013	0	3.3e+03	258
163	06:17:56	0.0042	4.16	1.5e-10	3.1e-11	9.1e-05	8e-07	0.00016	0	7e+03	250
164	06:18:10	0.00413	4.13	5.57e-10	4.7e-10	0.00012	1e-05	0.00018	0	7.5e+03	240
165	06:18:25	0.00405	4.09	3.71e-09	6.6e-10	0.00015	3e-06	0.00016	0	3.5e+03	264
166	06:18:41	0.00398	4.05	3.17e-10	1.7e-10	0.00013	9e-06	0.00019	0	1.7e+04	258
167	06:18:56	0.00391	4.02	2.67e-10	6.4e-11	0.00014	7e-06	0.00018	0	5.3e+03	245
168	06:19:11	0.00384	3.98	3.16e-10	4.5e-11	0.00014	5e-06	0.00019	0	5.4e+03	266
169	06:19:26	0.00376	3.94	1.39e-10	4.1e-11	0.00014	3e-06	0.0002	0	5.5e+03	245
170	06:19:41	0.00369	3.91	2.25e-10	1.8e-10	0.00012	2e-05	0.00017	4e-15	5.8e+03	348
171	06:22:14	0.00351	3.81	5.22e-10	1.5e-10	0.00011	2e-06	0.00013	0	7.8e+03	487
172	06:22:34	0.00344	3.77	4.89e-10	1.5e-10	0.00011	2e-06	0.00016	0	1.3e+04	335
173	06:22:54	0.00338	3.74	2.16e-10	4.4e-11	0.00015	7e-06	0.00028	0	3.6e+03	360
174	06:23:14	0.00331	3.7	2.78e-10	1.1e-10	0.00016	1e-05	0.00024	0	2.2e+03	332
175	06:23:34	0.00325	3.66	1.07e-09	9.8e-10	0.00019	1e-05	0.00022	0	4.2e+03	318
176	06:23:54	0.00318	3.63	1.49e-10	2.8e-11	0.00016	8e-06	0.00025	0	6.3e+03	283
177	06:24:14	0.00312	3.59	2.83e-10	8.6e-11	0.00018	8e-06	0.00023	0	2.3e+03	354
178	06:24:34	0.00305	3.55	1.26e-10	4.6e-11	0.00015	1e-05	0.00025	0	2.6e+03	346
179	06:24:54	0.00299	3.51	2.57e-10	5.8e-11	0.00017	8e-06	0.00025	0	1.5e+03	496
180	06:25:41	0.00293	3.48	1.16e-10	3.2e-11	9.2e-05	5e-06	0.00015	0	8.8e+03	618
181	06:26:21	0.00287	3.44	4.31e-10	2.3e-10	0.00012	5e-06	0.00015	0	2.3e+04	689
182	06:27:01	0.00281	3.4	1.86e-10	6.5e-11	0.00014	6e-06	0.00019	0	3.8e+03	689

183	06:27:41	0.00274	3.37	1.72e-10	9.9e-11	0.00014	9e-06	0.0002	0	3.3e+03	675
184	06:28:21	0.00269	3.33	1.95e-10	4.6e-11	0.00015	5e-06	0.00019	9e-15	6.1e+03	657
185	06:29:01	0.00263	3.29	1.11e-10	1.7e-11	0.00013	3e-06	0.0002	0	3.2e+03	667
186	06:29:41	0.00257	3.26	3.19e-10	1.5e-10	0.00016	9e-06	0.0002	0	1.3e+04	670
187	06:30:21	0.00251	3.22	1.74e-10	4e-11	0.00016	3e-06	0.00023	0	4.6e+03	679
188	06:31:01	0.00245	3.18	1.35e-10	4.1e-11	0.00016	7e-06	0.00023	0	8.2e+03	673
189	06:31:41	0.0024	3.15	1.22e-10	2.6e-11	0.00015	5e-06	0.00022	0	5.5e+03	846
190	06:33:34	0.00234	3.11	6.61e-11	6.2e-12	0.00012	4e-06	0.00022	0	1.4e+04	605
191	06:34:14	0.00228	3.07	3.26e-11	6.1e-12	0.00012	5e-06	0.00021	0	1.2e+04	1615
192	06:36:44	0.00231	3.09	1.48e-09	7e-11	0.00014	1e-06	9.7e-05	9e-08	2.2e+03	1408
193	06:38:09	0.00241	3.16	8.62e-10	6.5e-11	0.0001	1e-06	6.5e-05	3e-13	1.4e+03	1428
194	06:39:40	0.00252	3.23	1.39e-09	1.3e-10	7.2e-05	6e-07	5.4e-05	4e-08	1.8e+03	1634
195	06:42:21	0.00287	3.44	6.96e-05	6.9e-05	6e-05	3e-07	7e-06	2e-08	9.3e+02	978
196	06:42:59	0.00351	3.8	0.000138	0.00011	7.2e-06	3e-08	8.9e-07	1e-08	1.5e+02	289
197	06:43:11	0.00422	4.17	4.33e-09	5.8e-10	9.3e-07	3e-08	4.4e-08	4e-12	3.1e+02	405
198	07:01:58	0.00605	5	1.59e-09	3.9e-10	2.7e-07	1e-08	4.4e-07	3e-17	2.2e+03	329
199	07:02:13	0.00587	4.93	1.3e-09	5.4e-10	3.5e-07	6e-09	5.5e-07	0	3.3e+03	248
200	07:02:28	0.0057	4.85	7.85e-10	2.7e-10	3.5e-07	9e-09	7.1e-07	0	3e+03	240
201	07:02:43	0.00552	4.78	2.1e-10	8.4e-11	3.7e-07	5e-08	9.7e-07	0	8.3e+03	234
202	07:02:58	0.00535	4.7	3.42e-10	6.7e-11	6.5e-07	3e-08	1.6e-06	0	3.7e+04	226
203	07:03:13	0.00518	4.63	7.85e-10	9.4e-10	1.1e-06	3e-07	2.2e-06	0	5.3e+04	206
204	07:03:28	0.00502	4.55	9.75e-11	1.4e-10	9e-07	5e-07	3.1e-06	0	1e+04	229
205	07:03:43	0.00486	4.48	1.24e-10	2.6e-10	1.8e-06	8e-07	4.9e-06	0	1.8e+04	231
206	07:03:58	0.0047	4.41	2.07e-11	6.9e-12	1.3e-06	0	6.8e-06	0	1.2e+05	229
207	07:04:13	0.00454	4.33	3.24e-11	9.2e-13	1.6e-06	0	5.7e-06	0	2.9e+04	883
208	07:05:23	0.00439	4.26	1.86e-10	3.3e-11	4.5e-06	9e-08	1.1e-05	0	3.1e+04	575
209	07:05:48	0.00424	4.19	9.81e-11	3e-11	5.5e-06	3e-07	1.3e-05	0	1.3e+04	403
210	07:06:13	0.00409	4.11	4.56e-11	7.4e-11	4.6e-06	2e-06	1.6e-05	0	3.6e+04	367
211	07:06:38	0.00394	4.04	2.73e-11	7.9e-13	4.6e-06	0	1.4e-05	0	3e+04	1212
212	07:08:11	0.0038	3.96	2.44e-10	3.1e-11	1e-05	2e-07	2.1e-05	0	2.2e+05	790
213	07:08:51	0.00365	3.89	2.06e-10	4.8e-11	1.6e-05	6e-07	3e-05	0	2.8e+04	662
214	07:09:31	0.00352	3.81	5.3e-11	6.6e-11	1.2e-05	4e-06	3.9e-05	0	2.5e+05	539
215	07:10:11	0.00338	3.74	2.75e-10	7.2e-11	2.6e-05	1e-06	4.4e-05	0	1.9e+05	645
216	07:10:51	0.00325	3.67	1.19e-10	6.2e-11	2.9e-05	2e-06	6.2e-05	0	4.6e+04	607
217	07:11:31	0.00312	3.59	1.71e-10	3.1e-11	3.5e-05	2e-06	7e-05	0	2.5e+04	618
218	07:12:11	0.00299	3.52	2.25e-10	2.9e-11	4.8e-05	9e-07	9e-05	0	4.4e+04	662
219	07:12:51	0.00287	3.44	1.79e-10	7.1e-11	6.2e-05	5e-06	0.00012	0	1.1e+05	627

220	07:13:31	0.00275	3.37	1.64e-10	3.3e-11	6.5e-05	3e-06	0.00013	0	1.2e+04	612
221	07:14:11	0.00263	3.3	1.16e-10	2.4e-11	7.1e-05	3e-06	0.00013	8e-18	3.7e+04	1155
222	07:15:34	0.00254	3.24	5.71e-10	1.4e-10	0.00011	3e-06	0.00014	0	2.7e+03	758
223	07:16:14	0.00248	3.2	4.57e-10	1.4e-10	0.00012	4e-06	0.00016	0	5.9e+03	627
224	07:16:54	0.00243	3.17	3.89e-10	8.8e-11	0.00013	4e-06	0.00016	0	5.6e+03	673
225	07:17:34	0.00237	3.13	1.51e-10	3.4e-11	0.00012	3e-06	0.00016	0	1.2e+04	1193
226	07:20:35	0.00231	3.09	1.96e-10	3.3e-11	0.00011	3e-06	0.00018	0	5.7e+03	597
227	07:21:27	0.0022	3.02	4.63e-10	1.8e-10	0.00019	8e-06	0.00022	0	2.9e+03	572
228	07:21:55	0.00215	2.98	1.16e-10	2.8e-11	0.00016	5e-06	0.00024	0	3.7e+03	593
229	07:22:32	0.0021	2.94	6.56e-11	1.8e-11	0.00014	8e-06	0.00022	0	1.4e+04	1173
230	07:23:57	0.0021	2.94	8.83e-10	1.5e-10	0.00017	2e-06	0.00013	1e-14	1.3e+03	1008
231	07:24:56	0.00215	2.98	1.35e-09	2e-10	0.00014	2e-06	0.00012	9e-14	1e+03	771
232	07:25:39	0.0022	3.01	9.51e-10	1.1e-10	0.00012	1e-06	9.9e-05	9e-11	9.9e+02	1281
233	07:27:15	0.00224	3.04	1.54e-09	4.9e-10	0.00011	2e-06	8.9e-05	3e-10	1.9e+03	1190
234	07:28:27	0.0024	3.15	8.94e-10	1.6e-11	9.4e-05	6e-07	1.7e-05	9e-08	7.5e+02	1296
235	07:29:58	0.00268	3.33	6.73e-06	0	3.8e-05	1e-06	1.5e-05	7e-08	5.9e+02	1537
236	07:31:48	0.00298	3.51	9.4e-05	0.00015	1.5e-05	8e-08	5.7e-06	7e-09	4.8e+02	972
237	07:32:49	0.0033	3.69	3.36e-05	0	6e-06	4e-08	2.2e-06	4e-09	4.7e+02	670
238	07:33:23	0.00364	3.88	1.5e-05	0	2.3e-06	9e-09	8.7e-07	2e-08	5.9e+02	719
239	07:34:08	0.00399	4.06	0.000134	0	8.9e-07	2e-09	4e-07	7e-10	8e+02	806

TABLE V: Fit results for each step.

Appendix D: R source code

```
### R code from vignette source 'LHCdiffMD.Rnw'

### CONTROL PARAMETERS ###

# perform data collection
# 0 = fresh data collection
# string = use existing data file
#DATA.SOURCE <- 0
DATA.SOURCE <- "LHCDiffusion2012.RData"

# perform data analysis
# 0 = fresh data analysis
# string = use existing results file
#RESULTS.SOURCE <- 0
RESULTS.SOURCE <- "Results_v2.RData"
STEPS.SUBSET <- 0
DEBUG <- FALSE
BOOTSTRAP <- TRUE # estimate parameter errors with bootstrapping technique

######

cornicina <- paste(rep("=", 60), collapse="")
message("\n\n", cornicina, "\n",
    "  Diffusion analysis of collimator scans\n",
    cornicina, "\n", date(), "\n\n")
message("Initializing...\n")
library(MASS); library(mgcv); library(nlme); library(RSEIS); library(xtable)
options(continue="  ")

# seed for random number generation
set.seed(12345)

# useful constants
std.lvl <- pnorm(1)-pnorm(-1) # standard confidence level

# utility functions
# identity
idem <- function(x){x}
```

```
# beam size calculation
sigma2 <- function(sigma1=0.5, dpp=1.5e-4,
               beta2=100, D2=1000,
               beta1=50, D1=600) {
    sigma2 <- sqrt(beta2/beta1*(sigma1^2-(D1*dpp)^2)+(D2*dpp)^2)
    return(as.numeric(sigma2))}
# downsample a large array for plotting purposes
# function returns ordered vector of indices
idwnsmpl <- function(x=runif(100), n=10){
  i <- sort(sample(x=1:length(x), size=n))
  return(i)
}
# span
span <- function(x) diff(range(x))
# error on the mean
emean <- function(x=runif(100)){
  emean <- mad(x)/sqrt(length(x)-1)
  return(emean)}
# monotonic smooth spline, see ?pcls in package mgcv
mono.smooth.spline <- function(x=1:10, y=-(1:10)^2+rnorm(10,0,10),
    w=rep(1, 10), xnew=seq(0, 12, by=0.2), plot=FALSE) {
  dat <- data.frame(x=x, y=y)
  f.ug <- gam( y ~ s(x, k=10, bs="cr"))
  sm <- smoothCon( s(x, k=10, bs="cr"), dat, knots=NULL )[[1]]
  F <- mono.con(sm$xp, up=FALSE) # decreasing
  G <- list(X=sm$X, C=matrix(0,0,0), sp=f.ug$sp, p=sm$xp, y=y, w=w)
  G$Ain <- F$A; G$bin <- F$b; G$S<-sm$S; G$off<-0
  p <- pcls(G)
  fv <- Predict.matrix(sm, data.frame(x=xnew))%*%p; fv <- fv[,1]
  if(plot){
    plot(x, y)
    lines(x, fitted(f.ug))
    lines(xnew, fv, col="red") }
  return(fv)
}

# Matlab color palette
matlab.colors <- colorRampPalette(
  c("#00007F", "blue", "#007FFF", "cyan", "#7FFF7F", "yellow",
    "#FF7F00", "red", "#7F0000"))
```

```
options(
    SweaveHooks=list(fig=function() { # common Sweave plot options
        wi<-6; he<-6; pt<-10; fo<-"Times"
        ps.options(width=wi, height=he, pointsize=pt, family=fo, bg="white")
        pdf.options(width=wi, height=he, pointsize=pt, family=fo, bg="white")
        par(oma=rep(0,4), mar=c(4.2,4.2,0.2,0.2)) }))

# to include a figure in LaTeX, as floating object or simple graphics
IncludePlotInLatex <- function(file="Rplots",
                               flo=TRUE,
                               cap="This is the caption.",
                               lab="fig:dummy_label"){
if(flo){
    fi <- paste("\n\\begin{figure}\n\\includegraphics{", file,
            "}\n\\caption{", cap,
            " \\label{", lab, "}}\n\\end{figure}\n", sep="")
} else {
    fi <- paste("\n\\begin{center}\n\\includegraphics{", file,
                "}\n\\end{center}\n", sep="") }
cat(fi) }


message("Defining model...\n")
DefineModel <- function() {
# collimator position vs. time, linear
Jc <<- function(t, Jci=0.04, Jcf=0.05, dt=1){
  Jc <- 0*t + Jci # initial value
  subset <- dt<=t
  Jc[subset] <- Jcf # final value
  subset <- 0<t & t<dt
  Jc[subset] <- Jci + (Jcf-Jci)*t[subset]/dt # collimator moving
  return(Jc)
}

# slope of distribution vs. time, linear
Ac <<- function(t, Ai=1, Af=0.9, dt=1){
  Ac <- 0*t + Ai # initial value
  subset <- dt<=t
  Ac[subset] <- Af # final value
```

```
  subset <- 0<t & t<dt
  Ac[subset] <- Ai + (Af-Ai)*t[subset]/dt # collimator moving
  return(Ac)
}

# initial, final, and asymptotic distributions, linear
flin <<- function(J, A=1, J0=0.04){
  f <- 0*J # default outside range
  J0 <- abs(J0) # define edge as positive
  subset <- abs(J)<J0 # triangular with cusp at J=0
  f[subset] <- A*(J0 - abs(J[subset])) # linear
  return(f) }

# evolution of the distribution function (der=0)
# or its derivative (der=1), analytical formulas
fs <<- function(J, t, Ai=1, Af=0.9, Jci=0.04, Jcf=0.05, D=1e-5, dt=1, der=0) {
  if(der!=0 & der!=1) stop("Derivative ('der') must be 0 or 1.")
  fs <- 0 # default outside collimators
  if(Jci>Jcf) IN <- TRUE else IN <- FALSE # inward or outward step?
  if(t<0){ # initial distribution
    if(der==0) fs <- flin(J, A=Ai, J0=Jci)
    if(der==1) fs <- -Ai
  } else{ # evolution
    Jco <- Jc(t, Jci=Jci, Jcf=Jcf, dt=dt) # collimator position
    Aco <- Ac(t, Ai=Ai, Af=Af, dt=dt) # slope of asymptotic solution
    if(0<=J & J<=Jco){ # inside collimators
      s <- sqrt(2*D*t) # Gaussian sigma
      # terms common to both inward and outward step
      if(der==0) fs.base <- -pnorm(-J/s) * (Ai*(Jci-J) - Aco*(Jco-J)) +
          pnorm((J-2*Jco)/s) * (Ai*(Jci+J-2*Jco) - Aco*(J-Jco)) +
          s/sqrt(2*pi) * (
            exp(-0.5*(J/s)^2) * (Aco-Ai) +
            (-1)*exp(-0.5*((2*Jco-J)/s)^2) * (Aco-Ai) )
      if(der==1) fs.base <- (Ai-Aco)*(pnorm(-J/s) + pnorm((J-2*Jco)/s)) +
          (Ai*Jci-Aco*Jco)/sqrt(2*pi)/s *
              (exp(-0.5*(J/s)^2)+exp(-0.5*((J-2*Jco)/s)^2))
      # add specific terms
      if(IN) { # inward step
        if(der==0) fs <- fs.base +
            (-1)*Ai*(Jci+J-2*Jco) +
```

```
            2 * pnorm((Jco-J)/s) * Ai * (Jci-Jco)
        if(der==1) fs <- fs.base +
            (-1)*Ai + 2*Ai*(Jco-Jci)*exp(-0.5*((Jco-J)/s)^2)/sqrt(2*pi)/s
      } else { # outward step
        if(der==0) fs <- fs.base +
          pnorm((Jci-J)/s) * Ai * (Jci-J) +
          (-1)*pnorm((Jci+J-2*Jco)/s) * Ai * (Jci+J-2*Jco) +
          s/sqrt(2*pi) * (
            exp(-0.5*((Jci-J)/s)^2) * Ai +
            (-1)*exp(-0.5*((Jci+J-2*Jco)/s)^2) * Ai )
        if(der==1) fs <- fs.base +
            (-1)*Ai*(pnorm((Jci-J)/s)+pnorm((Jci+J-2*Jco)/s))
      }
    }
  }
  return(fs)
}
fv <<- Vectorize(fs, vectorize.args=c("J","t")) # vectorize the function

# measured loss rate
# extension of Mess and Seidel, NIM A 351, 279 (1994)
# G. Stancari, arXiv:1108.5010 [physics.acc-ph]
#
# t - t0 = time relative to initial collimator movement (s)
# D = diffusion rate (um^2/s)
# K = loss monitor conversion factor (V / (particles/s))
# B = loss monitor background (V)
# Ai = initial slope of particle distribution
# Af = final slope of particle distribution
# Jci = initial collimator action (um)
# Jcf = final collimator action (um)
# dt = time to move collimators from Jci to Jcf (s)

# loss rate vs. time
# particle flux at the collimator L = -D * (df/dJ)|J=Jc
# with conversion constant k and background B
loss.rate <<- function(t, t0=0, Ai=1, Af=0.9, Jci=0.04, Jcf=0.05, D=1e-5, dt=1,
                       K=1, B=0) {
  t <- t-t0; D <- abs(D)
  L <- -Ai + 0*t # default before step
```

```
  subset <- t>0 # evolution
  s <- sqrt(2*D*t[subset]) # Gaussian sigmas
  Jco <- Jc(t[subset], Jci=Jci, Jcf=Jcf, dt=dt) # collimator positions
  Aco <- Ac(t[subset], Ai=Ai, Af=Af, dt=dt) # slopes
  if(Jci>Jcf) IN <- TRUE else IN <- FALSE # inward or outward?
  # common terms
  C <- 2/sqrt(2*pi)
  L[subset] <- 2*(Ai-Aco)*pnorm(-Jco/s) +
    C/s * (Ai*Jci-Aco*Jco) * exp(-0.5*(Jco*Jco/s/s))
  if(IN) {
      L[subset] <- L[subset] - Ai - C/s * Ai * (Jci-Jco)
  } else {
      L[subset] <- L[subset] - 2 * Ai * pnorm((Jci-Jco)/s) }
  return(-K*D*L+B)
}

}
DefineModel()


getOption("SweaveHooks")[["fig"]]()
# sample parameters
J1 <- 0.040; J2 <- 0.050 # initial and final collimator actions [um]
dt <- 1 # time to move collimator from Jci to Jcf [s]
A1 <- 1 # initial slope of distribution function [um^-2]
A2 <- A1*(J1/J2) # final slope
D <- 1e-5 # diffusion coefficient [um^2/s]
# choose times to plot
tscale.short <- abs(J1-J2)^2/D
tscale.long <- min(J1,J2)^2/D
tt <- c(seq(0, dt, length=5), # collimator moving
        c(3*dt, tscale.short, 2*tscale.short), # short term after
        c(5*tscale.short, tscale.long, 2*tscale.long)) # long term after
co <- gray(seq(0.6, 0, length=length(tt))) # define colors, grayscale
Jmin <- 0; Jmax <- max(J1,J2) # plot range
layout(matrix(1:2, nrow=1, ncol=2)) # plot layout
for(i in 1:2) {
  if(i==1){ # left plot, inward
    JJci<-J2; JJcf<-J1; Ai<-A2; Af<-A1}
  if(i==2){ # right plot, outward
```

```
    JJci<-J1; JJcf<-J2; Ai<-A1; Af<-A2}
  JJ <- Jc(tt, Jci=JJci, Jcf=JJcf, dt=dt) # calculate coll. pos.
  AA <- Ac(tt, Ai=Ai, Af=Af, dt=dt) # calculate slopes
  ylabel <- expression(paste("Distribution function [", mu*m^{-1}, "]", sep=""))
  if(i==2) ylabel <- ""
  plot(c(Jmin,Jmax),
     c(0, max(JJci*Ai,JJcf*Af)),
     type="n",
     xlab=expression(paste("Action [", mu, "m]", sep="")),
     ylab=ylabel)
abline(v=JJ, col=co, lwd=2) # plot collimator positions
for(i in seq(along=tt)){ # plot asymptotic functions
  curve(fv(J=x, t=tt[i], Ai=Ai, Af=Af, Jci=JJci, Jcf=JJcf, D=D, dt=dt),
        Jmin, Jmax, add=TRUE, col=co[i], lwd=2) }
legend("bottomleft", bty="n", # describe times for each curve
       title="Time [s]", title.col="black",
       legend=formatC(tt, drop0=TRUE), col=co, text.col=co, cex=0.8)
}


getOption("SweaveHooks")[["fig"]]()
# sample parameters
J1 <- 0.040; J2 <- 0.050 # initial and final collimator actions [um]
dt <- 1 # time to move collimator from Jci to Jcf [s]
D <- c(1e-6, 1e-5, 1e-4) # show the effect of different diffusion coeff.
K <- 1; B <- 0
tscale.short <- abs(J1-J2)^2/min(D) # estimate time scales
tscale.long <- min(J1,J2)^2/min(D)
tmax <- tscale.short/2
co <- gray(seq(0.5, 0, length=length(D))) # define colors, grayscale
Jmin <- 0; Jmax <- max(J1,J2) # plot range
layout(matrix(1:2, nrow=1, ncol=2)) # plot layout
for(i in 1:2) {
  if(i==1){ # left plot, inward
    Jci<-J2; Jcf<-J1
    ylabel <- "Loss rate [arb. units]"
    legend.pos <- "topright"}
  if(i==2){ # right plot, outward
    Jci<-J1; Jcf<-J2
    ylabel <- ""
```

```
    legend.pos <- "bottomright"}
  ADD <- FALSE
  for(j in 1:length(D)){
    Ai <- 1/D[j] # scale slope to keep steady-state rates constant
    Af <- Ai*Jci/Jcf
    if(j>1) ADD <- TRUE
    curve(loss.rate(t=x, Jci=Jci, Jcf=Jcf, Ai=Ai, Af=Af, D=D[j],
                    dt=dt, K=K, B=B), -0.1*tmax, tmax,
          xlab="Time [s]", ylab=ylabel, add=ADD, lwd=2, col=co[j])}
  if(i==1) legend(legend.pos, bty="n", # print diffusion rate for each curve
     title=expression(paste("Diffusion rate [", mu*m^2/s, "])),
     title.col="black",
     legend=formatC(D, format="E", di=1), col=co, text.col=co)
}


message("Defining constants...\n")
# analysis information
Ana.Info <- paste("Analyzed on", date(), "\n", R.version.string)
# times
tlabel <- "Local time [Europe/Zurich]"
Unix.epoch <- "1970-01-01 00:00 UTC"
CERN.tz <- "Europe/Zurich"
read.ctime <- function(t) as.POSIXct(t, origin=Unix.epoch, tz="UTC")
local.time <- function(t) as.POSIXlt(t, origin=Unix.epoch, tz=CERN.tz)
MD.date <- "2012.06.22"
Rubber.Stamp <- paste("LHC Halo Diffusion Study\n", MD.date, sep="")
# experiment times
ti.se <- as.POSIXct("2012-06-22 05:48", tz=CERN.tz) # separated beams
tf.se <- as.POSIXct("2012-06-22 06:47", tz=CERN.tz)
ti.co <- as.POSIXct("2012-06-22 06:58", tz=CERN.tz) # collisions
tf.co <- as.POSIXct("2012-06-22 07:42", tz=CERN.tz)
# physical
mp <- 0.93827231 # proton mass [GeV]
Ekin <- 4e3 # kinetic energy [GeV]
gammar <- (Ekin+mp)/mp # relativistic gamma
betar <- sqrt(1-1/gammar^2) # relativistic beta
emi.nom.norm <- 3.75 # nominal normalized emittance [um]
emi.nom.geom <- emi.nom.norm / betar / gammar # nominal geometrical [um]
dpp.nom <- 1.1e-4 # momentum spread
```

```
BLM.cal <- 1.25E12 # calibration of losses [p/Gy]
# collimators
CollNames <- c("TCP.D6L7.B1", "TCP.C6R7.B2") # names
coll.speed <- 2 # nominal collimator speed [mm/s]
# beam centers from March 2012 collimator alignment
B1.v.center <- 0.240 # vertical B1 center [mm]
B2.h.center <- 0.520 # horizontal B2 center [mm]


message("Making lattice table...\n")
# lattice functions at collimators of interest
loc.name <- c("TCP.B1.h", "TCP.B1.v", "TCP.B2.h", "TCP.B2.v")
loc.s <- c(19791.2, 19789.2, 6461.7, 6459.7) # [m]
beta.x <- c(150.5, 158.8, 150.4, 158.7)
beta.y <- c(82.8, 78.3, 82.75, 78.3)
disp.x <- c(0.33, 0.34, -0.14, -0.15)
disp.y <- rep(0, 4)
sigma.x.nom <- sqrt(beta.x*emi.nom.geom + (disp.x*1e3*dpp.nom)^2) # [mm]
sigma.y.nom <- sqrt(beta.y*emi.nom.geom + (disp.y*1e3*dpp.nom)^2) # [mm]
Lattice.Table <- data.frame(loc.s, beta.x, beta.y, disp.x, disp.y,
                    sigma.x.nom, sigma.y.nom)
rownames(Lattice.Table) <- loc.name
colnames(Lattice.Table) <- c("s", "beta.x", "beta.y", "D.x",
                             "D.y", "sigma.x.nom", "sigma.y.nom")
Table <- Lattice.Table
colnames(Table) <- c("\\multicolumn{1}{c}{s~[m]}",
                     "\\multicolumn{1}{c}{$\\beta_x$~[m]}",
                     "\\multicolumn{1}{c}{$\\beta_y$~[m]}",
                     "\\multicolumn{1}{c}{$D_x$~[m]}",
                     "\\multicolumn{1}{c}{$D_y$~[m]}",
                     "\\multicolumn{1}{c}{$\\sigma_x^\\mathrm{nom}$~[mm]}",
                     "\\multicolumn{1}{c}{$\\sigma_y^\\mathrm{nom}$~[mm]}")
T <- xtable(Table,
            caption="Lattice functions and nominal beam sizes
                     at the collimators of interest.",
            label="tab:lattice_functions",
            align="l.......",
            digits=c(NA, rep(1, 3), rep(2, 2), rep(3, 2)),
            display=c("s", rep("f", 7)) )
print(T, floating.environment="table*",
```

```
      latex.environments=c("ruledtabular"), size="small",
      hline.after=0, sanitize.text.function=idem)


VarSuffix <- c("1.h", "1.v", "2.h", "2.v")
if(is.character(DATA.SOURCE)){
  message("Reading previously-saved data frame...")
  load(DATA.SOURCE)
} else {
  message("Reading raw data files...")
  InputVarNames <- character()} # list of variables to be saved


if(DATA.SOURCE==0){
message("Reading intensities...")
# file names
data.dir <- "../procdata/"
sets <- c(rep("1s", 2), rep("20ms", 2))
beams <- rep(c("B1", "B2"), 2)
fnames <- paste(data.dir, "intensity_", sets, "_", beams, ".txt.gz", sep="")
var.names <- paste("I", c(rep("s", 2), rep("f", 2)), ".", beams, sep="")
nco <- c(rep(15000, 2), rep(1e6, 2)) # approx. # of rows
# variable names for time stamps and intensities
tvar.names <- paste("t", var.names, sep="")
# time stamps are in ms (1-s data) or ns (20-ms data)
tconst <- c(rep(1e3, 2), rep(1e9, 2))
for(i in seq(along=fnames)){
  A <- read.table(fnames[i], colClasses="numeric", nrow=nco[i])
  assign(tvar.names[i], read.ctime(A$V1/tconst[i]), envir=.GlobalEnv)
  assign(var.names[i], A$V2, envir=.GlobalEnv) }
rm(A)
InputVarNames <- c(InputVarNames, tvar.names, var.names)
}


if(DATA.SOURCE==0){
message("Reading wire-scan emittances...")
# file names
fname <- "../rawdata/wire_scan_emittances.txt"
# read wire-scan data
```

```
EMIT.WS <- read.table(file=fname, header=TRUE,
             colClasses=c(rep("integer", 2), rep("numeric", 8), "character"))
temit.ws <- paste(MD.date, " ", EMIT.WS$HH, ":", EMIT.WS$MM, " ",
                  CERN.tz, sep="")
temit.ws <- as.POSIXct(temit.ws, origin=Unix.epoch, tz=CERN.tz)
# average inward and outward wire fly
emit.ws.B1.h <- apply(EMIT.WS[, 3:4], 1, mean)
emit.ws.B1.v <- apply(EMIT.WS[, 5:6], 1, mean)
emit.ws.B2.h <- apply(EMIT.WS[, 7:8], 1, mean)
emit.ws.B2.v <- apply(EMIT.WS[,9:10], 1, mean)

InputVarNames <- c(InputVarNames, "EMIT.WS", "temit.ws",
                   paste("emit.ws.B", VarSuffix, sep=""))
}



if(DATA.SOURCE==0){
message("Reading synchrotron-light emittances...")
# read synchrotron-light data
BSRT.B1.H <- read.table(file="../rawdata/BSRT_B1_H.txt.gz", header=TRUE,
                        colClasses=c("integer", NA, "numeric"))
BSRT.B1.V <- read.table(file="../rawdata/BSRT_B1_V.txt.gz", header=TRUE,
                        colClasses=c("integer", NA, "numeric"))
BSRT.B2.H <- read.table(file="../rawdata/BSRT_B2_H.txt.gz", header=TRUE,
                        colClasses=c("integer", NA, "numeric"))
BSRT.B2.V <- read.table(file="../rawdata/BSRT_B2_V.txt.gz", header=TRUE,
                        colClasses=c("integer", NA, "numeric"))
temit.sl.B1.h <- read.ctime(BSRT.B1.H$Timestamp)
temit.sl.B1.v <- read.ctime(BSRT.B1.V$Timestamp)
temit.sl.B2.h <- read.ctime(BSRT.B2.H$Timestamp)
temit.sl.B2.v <- read.ctime(BSRT.B2.V$Timestamp)
emit.sl.B1.h <- BSRT.B1.H$Emittance
emit.sl.B1.v <- BSRT.B1.V$Emittance
emit.sl.B2.h <- BSRT.B2.H$Emittance
emit.sl.B2.v <- BSRT.B2.V$Emittance
rm(BSRT.B1.H, BSRT.B1.V, BSRT.B2.H, BSRT.B2.V)
InputVarNames <- c(InputVarNames,
                   paste("temit.sl.B", VarSuffix, sep=""),
                   paste("emit.sl.B", VarSuffix, sep="") )
}
```

```
if(DATA.SOURCE==0){
message("Reading collimator positions...")
CP <- read.table("../rawdata/coll_pos.tsv.gz", colClasses=rep("numeric", 2),
                 nrow=240000)
var.names <- c("TCP.C6R7.B2:MEAS_LVDT_LD",
               "TCP.C6R7.B2:MEAS_LVDT_LU",
               "TCP.C6R7.B2:MEAS_LVDT_RD",
               "TCP.C6R7.B2:MEAS_LVDT_RU",
               "TCP.C6R7.B2:MEAS_MOTOR_LD",
               "TCP.C6R7.B2:MEAS_MOTOR_LU",
               "TCP.C6R7.B2:MEAS_MOTOR_RD",
               "TCP.C6R7.B2:MEAS_MOTOR_RU",
               "TCP.D6L7.B1:MEAS_LVDT_LD",
               "TCP.D6L7.B1:MEAS_LVDT_LU",
               "TCP.D6L7.B1:MEAS_LVDT_RD",
               "TCP.D6L7.B1:MEAS_LVDT_RU",
               "TCP.D6L7.B1:MEAS_MOTOR_LD",
               "TCP.D6L7.B1:MEAS_MOTOR_LU",
               "TCP.D6L7.B1:MEAS_MOTOR_RD",
               "TCP.D6L7.B1:MEAS_MOTOR_RU")
nvar <- length(var.names)
npt <- c(rep(14400, nvar/2), rep(14397, nvar/2)) # number of data points per variable
ji <- c(1, cumsum(npt)[1:(nvar-1)]+1) # initial and final indices for each variable
jf <- ji+npt-1
# average upstream and downstream jaw, both times and positions
t.CP.B2H <- read.ctime(0.5*(CP$V1[ji[1]:jf[1]]+CP$V1[ji[2]:jf[2]])/1e3)
t.CP.B1V <- read.ctime(0.5*(CP$V1[ji[9]:jf[9]]+CP$V1[ji[10]:jf[10]])/1e3)
CP.B2H <- 0.5*(CP$V2[ji[1]:jf[1]]+CP$V2[ji[2]:jf[2]])
CP.B1V <- 0.5*(CP$V2[ji[9]:jf[9]]+CP$V2[ji[10]:jf[10]])
rm(CP)
InputVarNames <- c(InputVarNames, "t.CP.B1V", "t.CP.B2H", "CP.B1V", "CP.B2H")
}


Nr <- 3750; Nc <- 3910 # number of rows and columns in GV's files
# losses at closest BLM and at all nearby BLMs
iBLM.B1V <- c(2811, 2809, 2807, 2805) # columns in GV's file
iBLM.B2H <- c(2970, 2972, 2974, 2976)
```

```
BLMNames <- c("BLMEI.06L7.B1E10 TCP.[A,B,C,D]6L7.B1",
              "BLMEI.06R7.B2I10 TCP.[A,B,C,D]6R7.B1")
if(DATA.SOURCE==0){
message("Reading fast losses...")
fpath <- "../rawdata/fast_BLMs/"
flist <- dir(path=fpath, pattern="*.txt.gz")
t.LO <- numeric() # time stamp
LO.B1V.A <- LO.B1V.B <- LO.B1V.C <- LO.B1V.D <- numeric()
LO.B2H.A <- LO.B2H.B <- LO.B2H.C <- LO.B2H.D <- numeric()
LO.B1V.TOT <- LO.B2H.TOT <- LO.ALL <- numeric()
for(fn in flist){
  message("\t", fn)
#  A <- read.table(file=paste(fpath, fn, sep=""), skip=2,
#                  colClasses=cCla, flush=TRUE, nrow=3800)
  A <- matrix(scan(file=paste(fpath, fn, sep=""), skip=2, n=Nr*Nc),
              nrow=Nr, ncol=Nc, byrow=TRUE)
  t.LO <- c(t.LO, A[, 2])
  LO.B1V.A <- c(LO.B1V.A, A[, iBLM.B1V[1]])
  LO.B1V.B <- c(LO.B1V.B, A[, iBLM.B1V[2]])
  LO.B1V.C <- c(LO.B1V.C, A[, iBLM.B1V[3]])
  LO.B1V.D <- c(LO.B1V.D, A[, iBLM.B1V[4]])
  LO.B2H.A <- c(LO.B2H.A, A[, iBLM.B2H[1]])
  LO.B2H.B <- c(LO.B2H.B, A[, iBLM.B2H[2]])
  LO.B2H.C <- c(LO.B2H.C, A[, iBLM.B2H[3]])
  LO.B2H.D <- c(LO.B2H.D, A[, iBLM.B2H[4]])
  LO.B1V.TOT <- c(LO.B1V.TOT, apply(A[ , iBLM.B1V], 1, sum))
  LO.B2H.TOT <- c(LO.B2H.TOT, apply(A[ , iBLM.B2H], 1, sum))
  LO.ALL <- c(LO.ALL, rowSums(A[, 3:Nc]))
}
rm(A)
t.LO <- read.ctime(t.LO/1e6)
InputVarNames <- c(InputVarNames, "t.LO", "LO.ALL",
                   "LO.B1V.TOT", "LO.B2H.TOT",
                   "LO.B1V.A", "LO.B1V.B", "LO.B1V.C", "LO.B1V.D",
                   "LO.B2H.A", "LO.B2H.B", "LO.B2H.C", "LO.B2H.D"
                   )
}


# save data if fresh
```

```
if(DATA.SOURCE==0){
  message("Saving input data frame...")
  message("Variable names: ", InputVarNames)
  save(list=InputVarNames, file="tmp_LHCDiffusion2012.RData") }


if(is.character(RESULTS.SOURCE)){
  message("Reading saved results from file ", RESULTS.SOURCE)
  load(RESULTS.SOURCE)
} else {
  message("Performing fresh data analysis...")
  OutputVars <- character() } # list of variables to save


getOption("SweaveHooks")[["fig"]]()
message("Making intensity plots...")
par(mar=rep(0,4), oma=c(3.2, 3.2, 0.2, 3.2), mgp=c(2, 0.5, 0), tcl=-0.25)
layout(matrix(1:2, nrow=2, ncol=1))

tl <- as.POSIXct(paste("2012-06-22", c("05:00", "08:00"), "CEST"),
                 origin=Unix.epoch, tz=CERN.tz) # time limits
Iu <- 1e11 # intensity units
yl <- c(0, 1.5) # intensity limits
cpl <- c(0, 5) # collimator position limits [mm]
Dev <- list(
  col = c("darkblue", "blue", "cyan", "darkred", "red", "magenta"),
  nam = c("B1 intensity (1 s)", "B1 intensity (20 ms)", "TCP.D6L7.B1",
          "B2 intensity (1 s)", "B2 intensity (20 ms)", "TCP.C6R7.B2")
)
npt <- 2048
# Beam 1
i <- idwnsmpl(tIs.B1, npt)
t <- local.time(tIs.B1[i])
y <- Is.B1[i]/Iu
plot(t, y, xlim=tl, ylim=yl, type="n", axes=FALSE)
# experiment times
tnum <- as.numeric(c(ti.se, tf.se, ti.co, tf.co))
abline(v=tnum, col="gray")
text(mean(tnum[1:2]), min(y), "Separated")
text(mean(tnum[3:4]), min(y), "Colliding")
```

```
points(t, y, type="l", col=Dev$col[1]) # plot intensities above lines
i <- idwnsmpl(tIf.B1, npt)
t <- local.time(tIf.B1[i])
y <- If.B1[i]/Iu
points(t, y, type="l", col=Dev$col[2])
axis(2, col=Dev$col[2], col.axis=Dev$col[2])
# collimator position
i <- idwnsmpl(t.CP.B1V, npt)
t <- local.time(t.CP.B1V[i])
y <- CP.B1V[i]
plot.window(xlim=tl, ylim=cpl)
points(t, y, type="l", col=Dev$col[3])
axis(4, col=Dev$col[3], col.axis=Dev$col[3])
# legend
box()
legend("topright", bty="n", legend=Rubber.Stamp)
legend("bottomleft", bty="n", legend=Dev$nam[1:3], text.col=Dev$col[1:3])

# Beam 2
i <- idwnsmpl(tIs.B2, npt)
t <- local.time(tIs.B2[i])
y <- Is.B2[i]/Iu
plot(t, y, xlim=tl, type="n", yaxt="n")
# experiment times
abline(v=tnum, col="gray")
points(t, y, type="l", col=Dev$col[4]) # plot intensities above lines
i <- idwnsmpl(tIf.B2, npt)
t <- local.time(tIf.B2[i])
y <- If.B2[i]/Iu
points(t, y, type="l", col=Dev$col[5])
axis(2, col=Dev$col[5], col.axis=Dev$col[5])
# collimator position
i <- idwnsmpl(t.CP.B2H, npt)
t <- local.time(t.CP.B2H[i])
y <- CP.B2H[i]
colo <- Dev$col[6]
plot.window(xlim=tl, ylim=cpl)
points(t, y, type="l", col=colo)
axis(4, col=colo, col.axis=colo)
# legend
```

```
#axis(1)
box()
legend("bottomleft", bty="n", legend=Dev$nam[4:6], text.col=Dev$col[4:6])

# axis labels
layout(1)
mtext(tlabel, 1, 2, outer=TRUE)
mtext(paste("Beam intensity [", as.character(Iu), "]", sep=""),
      2, 2, outer=TRUE)
mtext("Collimator position [mm]", 4, 2, outer=TRUE)



getOption("SweaveHooks")[["fig"]]()
message("Overview of collimator scan (separated beams)...")
tl <- c(ti.se, tf.se) # time limits

CollScanPlot <- function(){
par(mar=rep(0,4), oma=c(3.2, 3.2, 0.2, 3.2), mgp=c(2, 0.5, 0), tcl=-0.25)
layout(matrix(1:2, nrow=2, ncol=1))
yl <- c(1e-7, 1e-2) # loss limits
cpl <- c(0.5, 3.5) # collimator position limits [mm]
Dev <- list(
  col = c("blue", "cyan", "red", "magenta"),
  nam = c("Local losses (B1)", "TCP.D6L7.B1",
          "Local losses (B2)", "TCP.C6R7.B2")
)
npt <- 8192
i <- idwnsmpl(t.LO, npt)
# Beam 1 losses
t <- local.time(t.LO[i])
y <- LO.B1V.TOT[i]
plot(t, y, xlim=tl, ylim=yl, type="l", axes=FALSE, log="y", col=Dev$col[1])
axis(2, col=Dev$col[1], col.axis=Dev$col[1])
# collimator position
t <- local.time(t.CP.B1V)
y <- CP.B1V
plot.window(xlim=tl, ylim=cpl)
points(t, y, type="l", col=Dev$col[2])
axis(4, col=Dev$col[2], col.axis=Dev$col[2])
# legend
```

```
box()
legend("topright", bty="n", legend=Rubber.Stamp)
legend("bottomleft", bty="n", legend=Dev$nam[1:2], text.col=Dev$col[1:2])

# Beam 2 losses
t <- local.time(t.LO[i])
y <- LO.B2H.TOT[i]
plot(t, y, xlim=tl, ylim=yl, type="l", yaxt="n", log="y", col=Dev$col[3])
axis(2, col=Dev$col[3], col.axis=Dev$col[3])
# collimator position
t <- local.time(t.CP.B2H)
y <- CP.B2H
colo <- Dev$col[4]
plot.window(xlim=tl, ylim=cpl)
points(t, y, type="l", col=colo)
axis(4, col=colo, col.axis=colo)
# legend
box()
legend("bottomleft", bty="n", legend=Dev$nam[3:4], text.col=Dev$col[3:4])

# axis labels
layout(1)
mtext(tlabel, 1, 2, outer=TRUE)
mtext("Local losses [Gy/s]", 2, 2, outer=TRUE)
mtext("Collimator position [mm]", 4, 2, outer=TRUE)
}
CollScanPlot()


getOption("SweaveHooks")[["fig"]]()
message("Overview of collimator scan (colliding beams)...")
tl <- c(ti.co, tf.co) # time limits
CollScanPlot()


getOption("SweaveHooks")[["fig"]]()
message("Response of local loss rates...")
tl <- as.POSIXct(paste("2012-06-22", c("07:18", "07:30"), "CEST"),
                 origin=Unix.epoch, tz=CERN.tz) # time limits
```

```
par(mar=rep(0,4), oma=c(3.2, 3.2, 0.2, 3.2), mgp=c(2, 0.5, 0), tcl=-0.25)
layout(matrix(1:2, nrow=2, ncol=1))
yl <- c(1e-8, 1e-2) # loss limits
cpl <- c(0.2, 2) # collimator position limits [mm]
Dev <- list(
  vn = c(paste("LO.B1V.", LETTERS[1:4], sep=""), "",
         paste("LO.B2H.", LETTERS[1:4], sep=""), ""),
  col = c(topo.colors(4), "cyan", matlab.colors(4), "magenta"),
  nam = c(paste("BLM ", LETTERS[1:4], "6L7", sep=""), "TCP.D6L7.B1",
          paste("BLM ", LETTERS[1:4], "6R7", sep=""), "TCP.C6R7.B2")
)
npt <- 4096
# Beam 1 losses
plot(c(1,2), c(1,2), xlim=tl, ylim=yl, type="n", axes=FALSE, log="y")
cut <- tl[1] <= t.LO & t.LO <= tl[2]
i <- idwnsmpl(t.LO[cut], npt)
for(j in 1:4){
  t <- local.time(t.LO[cut][i])
  y <- get(Dev$vn[j])[cut][i]
  points(t, y, type="l", col=Dev$col[j])}
axis(2)
# collimator position
cp.cut <- tl[1] <= t.CP.B1V & t.CP.B1V <= tl[2]
t <- local.time(t.CP.B1V[cp.cut])
y <- CP.B1V[cp.cut]
plot.window(xlim=tl, ylim=range(y))
points(t, y, type="l", col=Dev$col[5])
axis(4, col=Dev$col[5], col.axis=Dev$col[5])
# legend
box()
legend("topright", bty="n", legend=Rubber.Stamp)
legend("bottomleft", bty="n", legend=Dev$nam[1:5], text.col=Dev$col[1:5])

# Beam 2 losses
plot(local.time(t.LO[cut]), LO.B2H.TOT[cut],
     xlim=tl, ylim=yl, type="n", yaxt="n", log="y")
for(j in 6:9){
  t <- local.time(t.LO[cut][i])
  y <- get(Dev$vn[j])[cut][i]
  points(t, y, type="l", col=Dev$col[j])}
```

```
axis(2)
# collimator position
cp.cut <- tl[1] <= t.CP.B2H & t.CP.B2H <= tl[2]
t <- local.time(t.CP.B2H[cp.cut])
y <- CP.B2H[cp.cut]
colo <- Dev$col[10]
plot.window(xlim=tl, ylim=range(y))
points(t, y, type="l", col=colo)
axis(4, col=colo, col.axis=colo)
# legend
box()
legend("bottomleft", bty="n", legend=Dev$nam[6:10], text.col=Dev$col[6:10])

# axis labels
layout(1)
mtext(paste("Local time [mm:ss] (t0 = ", tl[1], ")", sep=""), 1, 2, outer=TRUE)
mtext("Local losses [Gy/s]", 2, 2, outer=TRUE)
mtext("Collimator position [mm]", 4, 2, outer=TRUE)


if(RESULTS.SOURCE==0){
message("Calculating background losses...")
# "quiet time"
t1.bkg <- as.POSIXct("2012-06-22 05:28 CEST", origin=Unix.epoch, tz=CERN.tz)
t2.bkg <- as.POSIXct("2012-06-22 05:48 CEST", origin=Unix.epoch, tz=CERN.tz)
quiet <- t1.bkg <= t.LO & t.LO <= t2.bkg
Bg.B1V <- median(LO.B1V.TOT[quiet])
Bg.B2H <- median(LO.B2H.TOT[quiet])
Bg.ALL <- median(LO.ALL[quiet])
OutputVars <- c(OutputVars, "t1.bkg", "t2.bkg", "Bg.B1V", "Bg.B2H", "Bg.ALL")
message("Saving results...")
message("List of variables: ", paste(OutputVars, collapse=" "))
save(list=OutputVars, file="tmp_Results.RData")
}


if(RESULTS.SOURCE==0){
message("Intensity and BLM calibration...")
# calibration of loss rates vs. collimator position
# integral of losses = k * (# of lost particles)
```

```
# smooth-spline fit of number of particles
tN.B1 <- tIf.B1
N.B1 <- If.B1
subset <- ti.se <= tN.B1 & tN.B1 <= tf.co
tN.B1 <- tN.B1[subset]; N.B1 <- N.B1[subset]
Nsm.win <- 51 # intensity smoothing window [points]
Nsm.B1 <- smooth.spline(tN.B1, runmed(N.B1, Nsm.win))
tN.B2 <- tIf.B2
N.B2 <- If.B2
subset <- ti.se <= tN.B2 & tN.B2 <= tf.co
tN.B2 <- tN.B2[subset]; N.B2 <- N.B2[subset]
Nsm.B2 <- smooth.spline(tN.B2, runmed(N.B2, Nsm.win))
# define time intervals
dt.cal <- 20
tbreaks.se <- seq(ti.se, tf.se, by=dt.cal)
tbreaks.co <- seq(ti.co, tf.co, by=dt.cal)
tbreaks <- c(tbreaks.se, tbreaks.co)
CO <- c( rep(0, length(tbreaks.se)-1), -1, # collisions?
         rep(1, length(tbreaks.co)-1) )
nbreaks <- length(tbreaks)
# average collimator half gap
tCf.B1 <- cut(t.CP.B1V, breaks=tbreaks)
tCf.B2 <- cut(t.CP.B2H, breaks=tbreaks)
CPm.B1 <- tapply(CP.B1V, tCf.B1, median) - B1.v.center
CPm.B2 <- tapply(CP.B2H, tCf.B2, median) - B2.h.center
# average BLM loss rate
subset <- ti.se <= t.LO & t.LO <= tf.co
tLf <- cut(t.LO[subset], breaks=tbreaks)
tm <- tapply(as.numeric(t.LO[subset]), tLf, mean)
#Lm.B1 <- tapply(runmed(LO.B1V.TOT[subset], 5), tLf, median)
#Lm.B2 <- tapply(runmed(LO.B2H.TOT[subset], 5), tLf, median)
#Lm <- tapply(runmed(LO.ALL[subset], 5), tLf, median)
Lm.B1 <- tapply(LO.B1V.TOT[subset]-Bg.B1V, tLf, mean)
Lm.B2 <- tapply(LO.B2H.TOT[subset]-Bg.B2H, tLf, mean)
Lm <- tapply(LO.ALL[subset]-Bg.ALL, tLf, mean)
# average particle loss rate
# unclass(time) because derivative requires division
tNf.B1 <- cut(tN.B1, breaks=tbreaks)
tNf.B2 <- cut(tN.B2, breaks=tbreaks)
Nm.B1 <- tapply(N.B1, tNf.B1, median)
```

```
Nm.B2 <- tapply(N.B2, tNf.B2, median)
dNm.B1 <- predict(Nsm.B1, x=unclass(tbreaks[2:nbreaks]))$y -
  predict(Nsm.B1, x=unclass(tbreaks[1:(nbreaks-1)]))$y
dNm.B2 <- predict(Nsm.B2, x=unclass(tbreaks[2:nbreaks]))$y -
  predict(Nsm.B2, x=unclass(tbreaks[1:(nbreaks-1)]))$y
edNm.B1 <- tapply(N.B1, tNf.B1, emean) * sqrt(2)
edNm.B2 <- tapply(N.B2, tNf.B2, emean) * sqrt(2)
lambda.B1 <- -dNm.B1 / dt.cal / Nm.B1
lambda.B2 <- -dNm.B2 / dt.cal / Nm.B1
elambda.B1 <- edNm.B1 / dt.cal / Nm.B1
elambda.B2 <- edNm.B2 / dt.cal / Nm.B2
# calibration factor
k.B1 <- abs(Lm.B1 * dt.cal / dNm.B1)
k.B2 <- abs(Lm.B2 * dt.cal / dNm.B2)
k.ALL <- abs(Lm * dt.cal / (dNm.B1+dNm.B2))
ek.B1 <- k.B1 * (edNm.B1/dNm.B1)
ek.B2 <- k.B2 * (edNm.B2/dNm.B2)
ek.ALL <- k.ALL * sqrt( edNm.B1^2 + edNm.B2^2 ) / (dNm.B1+dNm.B2)
# calibration vs. half gap, regrouped
cases <- c("B1 V se", "B1 V co", "B2 H se", "B2 H co")
hgap <- list() # half gap [mm]
kfac <- ekfac <- list() # calibration factor [Gy/p]
hgfit <- lkfit <- lksmf <- list() # smoothing function of log10(k) vs. half gap
length(hgap) <- length(kfac) <- length(ekfac) <- length(hgfit) <-
    length(lkfit) <- length(lksmf) <- length(cases)
names(hgap) <- names(kfac) <- names(ekfac) <- names(hgfit) <-
    names(lkfit) <- names(lksmf) <- cases
hg.group <- 0.04 # half-gap intervals [mm]
smsp.df <- 4 # number of degrees of freedom in the smooth spline interpolation
x <- c(CPm.B1, CPm.B2)
y <- c(k.B1, k.B2)
ey <- c(ek.B1, ek.B2)
BEAM1 <- c(rep(TRUE, length(CPm.B1)), rep(FALSE, length(CPm.B2)))
sub <- list( BEAM1 & CO==0, BEAM1 & CO==1, !BEAM1 & CO==0, !BEAM1 & CO==1 )
for(i in 1:length(sub)){
  xx <- x[ sub[[i]] ]; yy <- y[ sub[[i]] ]
  xf <- as.factor(round(xx/hg.group)*hg.group)
  hgap[[i]] <- tapply(xx, xf, mean)
  kfac[[i]] <- tapply(yy, xf, median)
  ekfac[[i]] <- tapply(yy, xf, mad)
```

```
  elk <- ekfac[[i]] / kfac[[i]] * 0.4342945 # log10(e)
  wei <- (elk)^(-2); wei[wei==Inf] <- min(wei[wei!=Inf])/8
  hgfit[[i]] <- seq(min(hgap[[i]]), max(hgap[[i]]), length=100)
  lksmf[[i]] <- smooth.spline(x=hgap[[i]], y=log10(kfac[[i]]),
                              w=wei, df=smsp.df)
  lkfit[[i]] <- predict(lksmf[[i]], x=hgfit[[i]])$y
#  lkfit[[i]] <- mono.smooth.spline(x=hgap[[i]], y=log10(kfac[[i]]), w=wei,
#                                   xnew=hgfit[[i]])
#  lksmf[[i]] <- splinefun(hgfit[[i]], lkfit[[i]], method="natural")
}
# variables to save
OutputVars <- c(OutputVars, "Nsm.B1", "Nsm.B2", "dt.cal", "tm", "CO",
                "CPm.B1", "CPm.B2", "Lm", "Lm.B1",
                "Lm.B2", "Nm.B1", "Nm.B2", "dNm.B1", "dNm.B2",
                "edNm.B1", "edNm.B2",
                "lambda.B1", "lambda.B2",
                "elambda.B1", "elambda.B2",
                "k.B1", "k.B2", "k.ALL",
                "ek.B1", "ek.B2", "ek.ALL",
                "hg.group", "hgap", "kfac", "ekfac",
                "smsp.df", "hgfit", "lkfit", "lksmf")
message("Saving results...")
message("List of variables: ", paste(OutputVars, collapse=" "))
save(list=OutputVars, file="tmp_Results.RData")
}


getOption("SweaveHooks")[["fig"]]()
message("Plotting losses vs. total particle loss rate...")
par(oma=rep(0,4), mar=c(3.2, 3.2, 0.5, 1), mgp=c(2, 0.5, 0), tcl=-0.25)
layout(matrix(1:2, nrow=1, ncol=2))
# total losses
x <- Lm
y <- -1*(dNm.B1 + dNm.B2)/dt.cal
ey <- sqrt(edNm.B1^2 + edNm.B2^2)/dt.cal
sy <- rep(1, length(x)); sy[CO==1] <- 16
plot(x, y, xlab="Total BLM loss rate [Gy/s]", ylab="Particles loss rate [p/s]",
     pch=sy, log="xy")
arrows(x, y-ey, x, y+ey, length=0.05, angle=90, code=3)
legend("topleft", bty="n", legend=Rubber.Stamp)
```

```
legend("bottom", bty="n", legend=c("Separated", "Colliding"), pch=c(1, 16))
# each beam separately
x <- c(Lm.B1, Lm.B2)
y <- -1*c(dNm.B1, dNm.B2)/dt.cal
ey <- c(edNm.B1, edNm.B2)/dt.cal
co <- c(rep("blue", length(Lm.B1)), rep("red", length(Lm.B2)))
sy <- rep(1, length(x)); sy[c(CO,CO)==1] <- 16
plot(x, y, xlab="Local loss rate [Gy/s]", ylab="Particle loss rate [p/s]",
     col=co, pch=sy, log="xy")
arrows(x, y-ey, x, y+ey, length=0.05, angle=90, code=3, col=co)
legend("topleft", bty="n", legend=Rubber.Stamp)
legend("bottomright", bty="n", legend=c("B1", "B2"), text.col=c("blue", "red"))
legend("bottom", bty="n", legend=c("Separated", "Colliding"), pch=c(1, 16))


message("Setting up BLM calibration plots...")
BLM.cal.plot <- function(lopt="y"){
  par(mar=rep(0, 4), oma=c(4.2, 4.2, 2, 2), mgp=c(2,0.5,0), tcl=-0.25)
  layout(matrix(1:4, nrow=2, ncol=2, byrow=FALSE))
  xl <- range(hgap); xx <- seq(xl[1], xl[2], length=100)
  yl <- range(kfac)
  legend.list <- c("B1 V separated", "B1 V colliding",
                   "B2 H separated", "B2 H colliding")
  tcol <- c("blue", "blue", "red", "red")
  sy <- c(1, 16, 1 , 16)
  for(i in 1:4){
    xm <- hgap[[i]]
    ym <- kfac[[i]]
    eym <- ekfac[[i]]
    fsm <- lksmf[[i]]
    plot(ym ~ xm, log=lopt, xlim=xl, ylim=yl, type="n", axes=FALSE)
    grid(lty="solid")
    xx <- hgfit[[i]]; yy <- 10^lkfit[[i]]
    points(xx, yy, type="l", col=tcol[i])
    points(xm, ym, col=tcol[i], pch=sy[i]) # data points
    arrows(xm, ym-eym, xm, ym+eym, length=0.05, angle=90, code=3, col=tcol[i])
    legend("topright", bty="n", legend=legend.list[i], text.col=tcol[i])
    if(i==1) { axis(2); axis(3) }
#    if(i==2) legend("bottomleft", bty="n", legend=Rubber.Stamp)
    if(i==4) { axis(1); axis(4) }
```

```
    box()
  }
  layout(1)
  mtext(Rubber.Stamp, side=1, line=2, adj=0, font=3, cex=0.8, outer=TRUE)
  mtext("Collimator half gap [mm]", 1, 3, outer=TRUE)
  mtext("Local BLM calibration [Gy/p]", 2, 3, outer=TRUE)
}


getOption("SweaveHooks")[["fig"]]()
message("Plotting loss calibration (log scale)...")
BLM.cal.plot(lopt="y")


getOption("SweaveHooks")[["fig"]]()
message("Plotting loss calibration (linear scale)...")
BLM.cal.plot(lopt="")


if(RESULTS.SOURCE==0){
message("Calculating emittances, beam sizes, and emittance growth rates...")
# before shaving, separated beams
t.emit.max <- as.POSIXct("2012-06-22 06:08", tz=CERN.tz)
# initial emittances, beam sizes, growth rates, and core diffusion rates
W <- eW <- W.ws <- eW.ws <- W.sl <- eW.sl <- gamma <- egamma <- Dp <-
    eDp <- numeric(4)
names(W) <- names(eW) <- names(W.ws) <- names(eW.ws) <- names(W.sl) <-
    names(eW.sl) <- names(gamma) <- names(egamma) <- names(Dp) <-
    c("B1.h", "B1.v", "B2.h", "B2.v")
mW <- list(); length(mW) <- 4 # emittance growth model
# variable names
ws.var.names <- paste("emit.ws.", c(rep("B1", 2), rep("B2", 2)), ".",
                    rep(c("h", "v"), 2), sep="")
sl.var.names <- paste("emit.sl.", c(rep("B1", 2), rep("B2", 2)), ".",
                    rep(c("h", "v"), 2), sep="")
tvar.names <- paste("t", sl.var.names, sep="")
for(i in 1:4){
  # wire scans
  W.ws[i] <- get(ws.var.names[i])[1]
  j <- 2*i+1; k<-j+1
```

```
  eW.ws[i] <- abs(EMIT.WS[1, j]-EMIT.WS[1, k]) / sqrt(12)
  # sync light
  t <- as.numeric(get(tvar.names[i]))
  y <- get(sl.var.names[i])
  cut <- t <= t.emit.max
  # average emittance and statistical error
  W.sl[i] <- median(y[cut])
  eW.sl[i] <- mad(y[cut])/sqrt(length(y[cut]))
  # emittance growth rate
  t <- t[cut]
  y <- y[cut]
  mW[[i]] <- nls(y ~ y0 * exp(g*(t-t[1])), model=TRUE,
            start=list(y0=median(y), g=IQR(y)/diff(range(t))),
            control=list(warnOnly=TRUE))
  R <- summary(mW[[i]])$coefficients
  gamma[i] <- R["g", 1]
  egamma[i] <- R["g", 2]
}

# average wire scan and sync light
W <- 0.5*(W.sl + W.ws)
eW <- sqrt( ((W.sl-W.ws))^2/12 + eW.sl^2 + eW.ws^2)
# beam sizes
sigma.B1.v <- sqrt(W["B1.v"]*Lattice.Table["TCP.B1.v", "beta.y"]/gammar/betar)
sigma.B2.h <- sqrt(W["B2.h"]*Lattice.Table["TCP.B2.h", "beta.x"]/gammar/betar)
# core diffusion (slope of D(J))
Dp <- gamma * W / gammar / betar
eDp <- Dp * sqrt( (egamma/gamma)^2 + (eW/W)^2 )
# list of variables to save
OutputVars <- c(OutputVars, "t.emit.max",
                "W.ws", "eW.ws", "W.sl", "eW.sl",
                "W", "eW", "sigma.B1.v", "sigma.B2.h",
                "mW", "gamma", "egamma", "Dp", "eDp")
message("Saving results...")
message("List of variables: ", paste(OutputVars, collapse=" "))
save(list=OutputVars, file="tmp_Results.RData")
}


message("Making wire scan emittance table...")
```

```
Table <- data.frame(emit.ws.B1.h, emit.ws.B1.v,
                    emit.ws.B2.h, emit.ws.B2.v,
                    EMIT.WS$DESCR)
rownames(Table) <- format(temit.ws, format="%H:%M")
colnames(Table) <- c(
  "\\multicolumn{1}{c}{$\\epsilon_x^{\\mathrm{B1}}~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\epsilon_y^{\\mathrm{B1}}~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\epsilon_x^{\\mathrm{B2}}~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\epsilon_y^{\\mathrm{B2}}~\\mathrm{[\\mu m]}$}",
  "Notes")
T <- xtable(Table,
            caption="Normalized, 1$\\sigma$ wire-scan emittances.",
            label="tab:emit.ws",
            align="c....l",
            digits=c(NA, rep(3, 4), NA),
            display=c("s", rep("fg", 4), "s"))
print(T, floating.environment="table*",
      latex.environments=c("ruledtabular"), size="small",
      hline.after=0, sanitize.text.function=idem)


message("Making emittance table...")
Table <- data.frame(W.ws, eW.ws, W.sl, eW.sl, W, eW, gamma, egamma, Dp, eDp)
rownames(Table) <- names(W)
colnames(Table) <- c(
  "\\multicolumn{1}{c}{$\\epsilon_{\\mathrm{ws}}}~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\delta\\epsilon_{\\mathrm{ws}}}~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\epsilon_{\\mathrm{sl}}}~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\delta\\epsilon_{\\mathrm{sl}}}~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\epsilon~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\delta\\epsilon~\\mathrm{[\\mu m]}$}",
  "\\multicolumn{1}{c}{$\\gamma~\\mathrm{[1/s]}$}",
  "\\multicolumn{1}{c}{$\\delta\\gamma~\\mathrm{[1/s]}$}",
  "\\multicolumn{1}{c}{$D'~\\mathrm{[\\mu m / s]}$}",
  "\\multicolumn{1}{c}{$\\delta D'~\\mathrm{[\\mu m / s]}$}" )
T <- xtable(Table,
            caption="Initial emittances (wire scan, synchrotron light, average),
                     emittance growth rates~$\\gamma$,
                     and core diffusion slopes~$D'$.",
            label="tab:emittances",
```

```
            align="l......rrrr",
            digits=c(NA, rep(c(3, 2), 5)),
            display=c("s", rep("fg", 6), rep("g", 4)) )
print(T, floating.environment="table*",
      latex.environments=c("ruledtabular"), size="small",
      hline.after=0, sanitize.text.function=idem)


getOption("SweaveHooks")[["fig"]]()
message("Making emittance plots...")
par(mar=rep(0,4), oma=c(3.2,3.2,0.2,0.2), mgp=c(2,0.5,0), tcl=-0.25)
layout(matrix(1:4, nrow=4, ncol=1))

tl <- as.POSIXct(paste("2012-06-22", c("05:30", "08:00"), "CEST"),
                 origin=Unix.epoch, tz=CERN.tz) # time limits
tnum <- as.numeric(c(ti.se, tf.se, ti.co, tf.co))
yl <- c(0, 4) # emittance range

PlotDsc <- c("B1H", "B1V", "B2H", "B2V")
T1 <- paste("temit.sl.B", VarSuffix, sep="")
T2 <- rep("temit.ws", 4)
Var1 <- paste("emit.sl.B", VarSuffix, sep="")
Var2 <- paste("emit.ws.B", VarSuffix, sep="")
sl.col <- "black"
ws.col <- "darkorange"

for(i in 1:4){
  t <- local.time(get(T1[i])) # sync-light
  y <- runmed(get(Var1[i]), 3)
  plot(t, y, xlim=tl, ylim=yl, xaxt=ifelse(i==4, "s", "n"), type="n")
  abline(v=tnum, col="gray")
  abline(v=t.emit.max, lty="dashed")
  points(mW[[i]]$model$t, predict(mW[[i]]), type="l", col="red")
  if(i==1){
    legend("topleft", bty="n", legend=Rubber.Stamp)
    legend("left", bty="n", legend=c("Synchrotron light", "Wire scan"),
           text.col=c(sl.col, ws.col))
    text(mean(tnum[1:2]), 0, "Separated")
    text(mean(tnum[3:4]), 0, "Colliding") }
  points(t, y, col=sl.col, type="l")
```

```
  t <- local.time(get(T2[i])) # wire scan
  y <- get(Var2[i])
  points(t, y, col=ws.col, pch=16, cex=4/3)
  legend("bottomleft", bty="n", legend=PlotDsc[i])
}
# axis labels
layout(1)
mtext(tlabel, 1, 2, outer=TRUE)
mtext(expression(paste("Normalized emittance (1", sigma, ") [", mu, "m]")),
      2, 2, outer=TRUE)


message("Finding and calculating collimator steps...")
READ.STEPS.FROM.FILE <- TRUE
# step definitions
Steps <- list(
              time=numeric(),
              time.local=character(),
              tstart=numeric(),
              tstop=numeric(),
              cpi=numeric(), # initial and final collimator positions [mm]
              cpf=numeric(),
              inw=logical(), # inward step?
              center=numeric(), # axis center
              hgi=numeric(), # half gaps before and after [mm]
              hgf=numeric(),
              size=numeric(), # step size [um]
              dtg=numeric(), # approximate time to make step [s]
              Jci=numeric(), # collimator actions before and after [um]
              Jcf=numeric(),
              deltaJ=numeric(),
              xs=numeric(),
              collisions=logical(),
              plane=character()
              )

# utility to select regions of interest for interpolation
SelectRegions <- function(
  fname="tmp.txt",
  t=seq(10, 14, by=0.1), # fake time and
```

```
  y=t, # loss data
  guesses=11:13, # initial guesses
  largew=2*60, # wide window to select initial, step, and final times
  smallw=2){ # narrow window to determine step time precisely
  guess1 <- guesses
  guess2 <- guesses
  t <- as.numeric(t)
  for(i in seq(along=guesses)){
    for( window in c(largew, smallw) ){ # wide and narrow time windows
      sub <- abs(t-guesses[i]) <= window & y>0
      plot(t[sub], y[sub], log="y", cex=0.8)
      rug(guesses[i], col="red")
      if(window==largew){
        guess1[i] <- locator(1)$x
        guesses[i] <- locator(1)$x
        guess2[i] <- locator(1)$x
      } else { guesses[i] <- locator(1)$x } } }
  GU <- cbind(guess1, guesses, guess2)
  colnames(GU) <- c("t1", "ts", "t2")
  write.table(GU, file=fname, row.names=FALSE)
  return(GU)
}

  step.thr <- 0.004 # step threshold definition [mm]
  # search intervals
  ti.B1.se <- as.POSIXct("2012-06-22 05:56:23", origin=Unix.epoch, tz=CERN.tz)
  tf.B1.se <- as.POSIXct("2012-06-22 06:41:29", origin=Unix.epoch, tz=CERN.tz)
  ti.B1.co <- as.POSIXct("2012-06-22 07:01:06", origin=Unix.epoch, tz=CERN.tz)
  tf.B1.co <- as.POSIXct("2012-06-22 07:34:57", origin=Unix.epoch, tz=CERN.tz)
  ti.B2.se <- as.POSIXct("2012-06-22 06:07:27", origin=Unix.epoch, tz=CERN.tz)
  tf.B2.se <- as.POSIXct("2012-06-22 06:43:34", origin=Unix.epoch, tz=CERN.tz)
  ti.B2.co <- as.POSIXct("2012-06-22 07:01:41", origin=Unix.epoch, tz=CERN.tz)
  tf.B2.co <- as.POSIXct("2012-06-22 07:34:43", origin=Unix.epoch, tz=CERN.tz)
  # find step indices
  step.indices <- function(CP=CP.B1V, thr=step.thr,
                           subset=rep(TRUE, length(CP))) {
      dcp <- c(0, diff(CP))
      moving <- start.moving <- rep(FALSE, length(CP))
      moving[ abs(dcp)>=thr ] <- TRUE
      start.moving[ c(0, diff(moving)) == 1 ] <- TRUE
```

```
      isteps <- which( start.moving & subset )
      di <- c(0, diff(isteps))
      not.too.close <- di>=4
      isteps <- isteps[not.too.close]
      return(isteps)
  }

# get steps from file or use automatic algorithm?
if(READ.STEPS.FROM.FILE){
  ST <- read.table("../rawdata/step_info.txt", header=TRUE,
                   colClasses=c( rep("numeric", 3), "logical", "character"))
  Steps$tstart <- ST$t.start
  Steps$time <- ST$t.step
  Steps$tstop <- ST$t.stop
  Steps$collisions <- ST$collisions
  Steps$plane <- ST$plane
  nvsteps <- length(Steps$time[Steps$plane=="B1V"])
  nhsteps <- length(Steps$time[Steps$plane=="B2H"])
} else {
  # B1, vertical
  SEP <- ti.B1.se <= t.CP.B1V & t.CP.B1V <= tf.B1.se
  COL <- ti.B1.co <= t.CP.B1V & t.CP.B1V <= tf.B1.co
  CUT <- SEP | COL
  istep.v <- step.indices(CP=CP.B1V, subset=CUT)
  tstep.v <- t.CP.B1V[istep.v]
  collisions.v <- COL[istep.v] # collisions or separated beams?
  plane.v <- rep("B1V", length(tstep.v))
  nvsteps <- length(tstep.v)
  # B2, horizontal
  SEP <- ti.B2.se <= t.CP.B2H & t.CP.B2H <= tf.B2.se
  COL <- ti.B2.co <= t.CP.B2H & t.CP.B2H <= tf.B2.co
  CUT <- SEP | COL
  istep.h <- step.indices(CP=CP.B2H, subset=CUT)
  tstep.h <- t.CP.B2H[istep.h]
  collisions.h <- COL[istep.h] # collisions or separated beams?
  plane.h <- rep("B2H", length(tstep.h))
  nhsteps <- length(tstep.h)
  # fill and save data frame
  Steps$time <- as.numeric(c(tstep.v, tstep.h))
  Steps$tstart <- Steps$time - 8 # [s]
```

```
  for(i in 1:nsteps){ # stop before next step or end of experiment
    later.steps.in.same.plane <-
      Steps$time>Steps$time[i] & Steps$plane==Steps$plane[i]
    Steps$tstop[i] <- min( Steps$time[later.steps.in.same.plane],
                           ifelse(Steps$time[i] > tf.se,
                                  as.numeric(tf.co),
                                  as.numeric(tf.se)) ) - 1 }
  Steps$collisions <- c(collisions.v, collisions.h)
  Steps$plane <- c(plane.v, plane.h)
  istep <- c(istep.v, istep.h)
}

nsteps <- length(Steps$time)
Steps$time.local <- format(local.time(Steps$time), "%H:%M:%S")
# refine fit intervals
REFINE.STEPS <- FALSE
if(REFINE.STEPS){
GU.v <- SelectRegions(t=as.numeric(t.LO), y=LO.B1V.TOT,
                      guesses=Steps$time[Steps$plane=="B1V"],
                      fname="tmp_guesses_v.txt")
GU.h <- SelectRegions(t=as.numeric(t.LO), y=LO.B2H.TOT,
                      guesses=Steps$time[Steps$plane=="B2H"],
                      fname="tmp_guesses_h.txt")
Steps$tstart <- c( GU.v["t1",], GU.h["t1",] )
Steps$time <- c( GU.v["ts",], GU.h["ts",] )
}
# other useful step properties
for(i in 1:nsteps){
  # collimator positions before and after
  if(Steps$plane[i]=="B1V"){
    tCP <- t.CP.B1V
    CP <- CP.B1V
  } else {
    tCP <- t.CP.B2H
    CP <- CP.B2H }
  before <- (Steps$time[i]-9) <= tCP & tCP <= (Steps$time[i]-2)
  after  <- (Steps$time[i]+2) <= tCP & tCP <= (Steps$time[i]+9)
  Steps$cpi[i] <- median(CP[before])
  Steps$cpf[i] <- median(CP[after])
}
```

```
# step sizes, half gaps, collimator actions
Steps$inw <- Steps$cpf < Steps$cpi
Steps$size <- (Steps$cpf-Steps$cpi)*1e3 # [um]
Steps$dtg <- abs(Steps$size)*1e-3 / coll.speed # [s]
Steps$center <- c( rep(B1.v.center, nvsteps), rep(B2.h.center, nhsteps) )
Steps$hgi <- Steps$cpi - Steps$center
Steps$hgf <- Steps$cpf - Steps$center
betas <- c( rep(Lattice.Table["TCP.B1.v", "beta.y"], nvsteps),
            rep(Lattice.Table["TCP.B2.h", "beta.x"], nhsteps) )
sigmas <- as.numeric(c( rep(sigma.B1.v, nvsteps), rep(sigma.B2.h, nhsteps) ))
Steps$Jci <- (Steps$hgi)^2 / 2 / betas
Steps$Jcf <- (Steps$hgf)^2 / 2 / betas
Steps$deltaJ <- abs(Steps$Jci-Steps$Jcf)
Steps$xs <- 0.5*(Steps$hgi+Steps$hgf) / sigmas # in units of sigmas
# save data frame
save(Steps, file="tmp_steps.RData")


if(RESULTS.SOURCE==0){
message("Analyzing collimator steps...")

# choose subset of steps to analyze
if(STEPS.SUBSET==0) explist <- 1:nsteps else explist <- STEPS.SUBSET
Kg <- 1 # calibration factor for losses, default [Gy/p]
Ksmi <- numeric(nsteps) # calibration factor, from smooth spline [Gy/p]
Ksmf <- numeric(nsteps) # initial (i) and final (f)
Bg <- numeric(nsteps) # background level
# preliminary measurements and guesses
Lbef <- numeric(nsteps) # loss level before
Laft <- numeric(nsteps) # loss level after
Lrmsbef <- numeric(nsteps) # fluctuations in losses before
Lrmsaft <- numeric(nsteps) # fluctuations in losses after
Dg.pk <- numeric(nsteps) # diffusion from peak [um^2/s]
Dg.tr <- numeric(nsteps) # diffusion from transient [um^2/s]
eDg.tr <- numeric(nsteps) # uncertainty on diffusion from transient [um^2/s]
Dg <- numeric(nsteps) # final guess for diffusion coefficient before interp. [um^2/s]
lossrange <- numeric(nsteps) # estimate of range of losses
DAig <- numeric(nsteps) # guess of initial steady-state loss level [particles/s]
DAfg <- numeric(nsteps) # guess of final steady-state loss level [particles/s]
# acceptable diffusion rates
```

```
Ddefault <- 1E-9
Dmin <- 1E-16
Dmax <- 1E4
lDrange <- 2 # +/- search range for log10(D)
# fit info and results
wwfit <- list(nsteps) # inverse square root fit: absissa
yyfit <- list(nsteps) # ordinate
mmtr <- list(nsteps) # linear model
tfit <- list(nsteps) # times [s]
yfit <- list(nsteps) # response [arb. units]
wfit <- list(nsteps) # weights
rfit <- list(nsteps) # residuals
dof <- numeric(nsteps) # degrees of freedom
chi2 <- numeric(nsteps) # chi-squared
fit <- list(nsteps) # model
conv <- logical(nsteps) # convergence?
hess <- list(nsteps) # Hessian matrix
errm <- list(nsteps) # covariance matrix
corm <- list(nsteps) # correlation matrix
pboot <- list(nsteps) # bootstrap estimate of parameter distribution
D <- numeric(nsteps) # diffusion rate [um^2/s]
eD <- numeric(nsteps) # estimate of standard error
DAi <- numeric(nsteps) # initial steady-state loss level [particles/s]
eDAi <- numeric(nsteps) # estimate of standard error
DAf <- numeric(nsteps) # final steady-state loss level [particles/s]
eDAf <- numeric(nsteps) # estimate of standard error
t0f <- numeric(nsteps) # fitted collimator movement time [s]
et0f <- numeric(nsteps) # estimate of standard error
dtf <- numeric(nsteps) # fitted time to move collimator [s]
edtf <- numeric(nsteps) # estimate of standard error

# analyze steps
for(ex in explist){
  message("\n\tAnalyzing step #", ex, "\n")
  # define variables of interest
  tcut <- Steps$tstart[ex]<=t.LO & t.LO<=Steps$tstop[ex]
  x <- as.numeric(t.LO[tcut]) - as.numeric(Steps$time[ex]) # time since step [s]
  if(Steps$plane[ex]=="B1V"){
    y<-LO.B1V.TOT[tcut]; Bg[ex] <- Bg.B1V
  } else {
```

```
    y<-LO.B2H.TOT[tcut]; Bg[ex] <- Bg.B2H }
  x <- x[y>0]; y <- y[y>0]
  # interpolated BLM calibration factor
  if(Steps$plane[ex]=="B1V" & !Steps$collisions[ex]) lkfu <- lksmf[["B1 V se"]]
  if(Steps$plane[ex]=="B1V" &  Steps$collisions[ex]) lkfu <- lksmf[["B1 V co"]]
  if(Steps$plane[ex]=="B2H" & !Steps$collisions[ex]) lkfu <- lksmf[["B2 H se"]]
  if(Steps$plane[ex]=="B2H" &  Steps$collisions[ex]) lkfu <- lksmf[["B2 H co"]]
  Ksmi[ex] <- predict(lkfu, x=Steps$hgi[ex])$y; Ksmi[ex] <- 10^(Ksmi[ex])
  Ksmf[ex] <- predict(lkfu, x=Steps$hgf[ex])$y; Ksmf[ex] <- 10^(Ksmf[ex])
  # define regions: before, transient, after
  t0bef <- -0.5; t0aft <- 4
  bef <- x < t0bef
  aft <- x > t0aft
  if(DEBUG) {dev.new(); plot(x, y)}
  # loss levels and fluctuations
  Lbef[ex] <- median(y[bef]-Bg[ex])
  Laft[ex] <- median(y[aft]-Bg[ex])
  if(abs(Steps$tstop[ex]-Steps$time[ex]) < t0aft) Laft[ex] <- Lbef[ex]
  Lrmsbef[ex] <- mad(y[bef])
  Lrmsaft[ex] <- mad(y[aft])
  if(DEBUG) message("Lbef = ", Lbef[ex], ", rms ", Lrmsbef[ex])
  if(DEBUG) message("Laft = ", Laft[ex], ", rms ", Lrmsaft[ex])
  # estimate peak or dip
  gt0 <- x>-0.5
#  lora <- quantile(Lo[gt0], probs=c(.01,.99), na.rm=TRUE)
  lora <- range(y[gt0])
  lossrange[ex] <- max(abs(lora-Lbef[ex]))
  lossrange[ex] <- abs(lossrange[ex] - abs(Lbef[ex]-Laft[ex]))
  if(DEBUG) message("Loss range = ", lossrange[ex])
  # diffusion rate estimate from peak/dip value
  Dg.pk[ex] <- Ddefault
  if( (Steps$dtg[ex]>0) & (Lbef[ex]>0) & (lossrange[ex]>0) )
    Dg.pk[ex] <- ( Steps$deltaJ[ex]*Lbef[ex]/lossrange[ex] )^2 /
                 pi / Steps$dtg[ex]
  if(DEBUG) message("Dg.pk = ", formatC(Dg.pk[ex], di=4))
  # diffusion rate estimate from simplfied model applied to transient
  Dg.tr[ex] <- Ddefault
  eDg.tr[ex] <- 0.1*Ddefault
  # define transient region
  ttra <- 3
```

```
  tra <- x>ttra
  xx <- x[tra]
  yyfit[[ex]] <- yy <- y[tra]
  if(Steps$inw[ex]) sgn<-+1 else sgn<--1
  wwfit[[ex]] <- ww <- Laft[ex] + Laft[ex]*sgn*Steps$deltaJ[ex]/sqrt(pi*xx)
#  ww <- Laft[ex]*sgn*deltaJ[ex]/sqrt(pi*xx)
  if(DEBUG) {dev.new(); plot(ww, yy)}
  mmtr[[ex]] <- mtr <- lm(yy ~ ww)
  inv.sqrt.DD <- coef(mtr)["ww"]
  ei <- summary(mtr)$coef[1,2]
#  ei <- diff(confint(mtr, parm="ww", level=std.lvl))/2
  if(DEBUG) message("Linear slope: ", inv.sqrt.DD, " +/-", ei)
  if(DEBUG) abline(mtr, col="red")
  if(inv.sqrt.DD!=0){
    Dg.tr[ex] <- 1/inv.sqrt.DD^2
    eDg.tr[ex] <- abs(0.5*ei/inv.sqrt.DD^1.5)}
  if(DEBUG) message("Dg.tr = ", formatC(Dg.tr[ex], di=4),
                    " +/- ", formatC(eDg.tr[ex], di=3))
  # diffusion rate guess
  D[ex] <- Dg[ex] <- Ddefault
  eD[ex] <- 0.1*Ddefault
  if( Dmin<=Dg.pk[ex] & Dg.pk[ex]<=Dmax &
      Dmin<=Dg.tr[ex] & Dg.tr[ex]<=Dmax   ){
    rD <- range(log10(c(Dg.pk[ex], Dg.tr[ex])))
#    D[ex] <- Dg[ex] <- 10^((.8)*rD[1] + (.2)*rD[2]) # more weight to small one
#    D[ex] <- Dg[ex] <- sqrt(Dg.pk[ex]*Dg.tr[ex]) # geometrical mean
    # trust transient more
#    D[ex] <- Dg[ex] <-
#        10^((0.2)*log10(Dg.pk[ex]) + (0.8)*log10(Dg.tr[ex]))
#    D[ex] <- Dg[ex] <- Dg.pk[ex]
    D[ex] <- Dg[ex] <- Dg.tr[ex]
#    D[ex] <- Dg[ex] <- 1E-7
#    D[ex] <- Dg[ex] <- mean(Dg.pk[ex], Dg.tr[ex])
#    D[ex] <- Dg[ex] <- min(Dg.pk[ex], Dg.tr[ex])
    eD[ex] <- abs(Dg.pk[ex]-Dg.tr[ex])/sqrt(12)}
  # guesses of slopes of distribution function
  DAi[ex] <- DAig[ex] <- abs(Lbef[ex])
  DAf[ex] <- DAfg[ex] <- abs(Laft[ex])
  if(Lbef[ex]>0)
    eDAi[ex] <- Lrmsbef[ex]*DAig[ex]/Lbef[ex] else eDAi[ex] <- 0.1*DAig[ex]
```

```
  if(Laft[ex]>0)
    eDAf[ex] <- Lrmsaft[ex]*DAfg[ex]/Laft[ex] else eDAf[ex] <- 0.1*DAfg[ex]
  # guesses for collimator step time and duration
  t0f[ex] <- 0
  et0f[ex] <- 1/12.5/sqrt(12)
  dtf[ex] <- Steps$dtg[ex]
  edtf[ex] <- 1/12.5/sqrt(12)
  # define weights
  wei <- rep(1/Lrmsbef[ex], length(y))
  rwei <- range(wei)
  if(DEBUG) message("Range of weights is [", formatC(rwei[1], di=3),
  ",", formatC(rwei[2], di=3), "]" )

  # minimize sum of deviations
  # need to check distribution of response (Poisson+oscillations?)
  fom <- function(p){ # figure-of-merit to minimize
    ye <- loss.rate(t=x, K=Kg, B=Bg[ex],
                    Jci=Steps$Jci[ex], Jcf=Steps$Jcf[ex],
                    D=10^(p[1]),
                    Ai=10^(p[2]-p[1]), Af=10^(p[3]-p[1]),
#                    t0=p[4], dt=dtf[ex])
                    t0=t0f[ex], dt=dtf[ex])
    fom <- sum(wei*abs(y-ye), na.rm=TRUE)
#    fom <- mean( log(1+0.5*(wei*(y-ye))^2) , na.rm=TRUE )
    return(fom)}
  chisqr <- function(p){ # chi-square for error estimation
    ye <- loss.rate(t=x, K=Kg, B=Bg[ex],
                    Jci=Steps$Jci[ex], Jcf=Steps$Jcf[ex],
                    D=10^(p[1]),
                    Ai=10^(p[2]-p[1]), Af=10^(p[3]-p[1]),
#                    t0=p[4], dt=dtf[ex])
                    t0=t0f[ex], dt=dtf[ex])
    chisqr <- sum(wei*wei*(y-ye)*(y-ye), na.rm=TRUE)
    return(chisqr)}
  # parameter definitions
  if(DEBUG) message("Initial guesses:\n\tD = ", Dg[ex],
                    "\n\tDAi = ", DAig[ex],
                    "\n\tDAf = ", DAfg[ex])
#                    "\n\tt0 = ", t0f[ex],
#                    "\n\tdt = ", dtf[ex])
```

```
  pnam <- c("lD", "lDAi", "lDAf")
  ipar <- c(log10(c(Dg[ex], DAig[ex], DAfg[ex]))) # initial values
  npar <- length(ipar)
  lop <- c(ipar[1]-lDrange, ipar[2]-0.5, ipar[3]-0.5) # lower limits
  upp <- c(ipar[1]+lDrange, ipar[2]+0.5, ipar[3]+0.5) # upper limits
  pars <- data.frame(initial=ipar, lower=lop, upper=upp,
                     fitted=ipar, errors=rep(0, npar)) # parameter summary
  rownames(pars) <- pnam
  # save model inputs
  tfit[[ex]] <- x
  yfit[[ex]] <- y
  wfit[[ex]] <- wei
  # function minimization
  if(DEBUG) message("Fitting with ", npar, " parameter(s)")
  fit[[ex]] <- optim(par=ipar, fn=fom, hessian=FALSE
#                , method="BFGS"
                , method="L-BFGS-B", lower=lop, upper=upp
#                , control=list(maxit=1e2, abstol=1e-2, reltol=1e-2)
               )
  pars$fitted <- pa <- fit[[ex]]$par
  pars$errors <- er <- rep(0, npar)
  chi2.properties <- fdHess(pa, chisqr)
  chi2[ex] <- chisqr(pa)
  dof[ex] <- length(x)-npar
#  hess[[ex]] <- H <- fit$hessian
  hess[[ex]] <- H <- chi2.properties$Hessian
  errm[[ex]] <- emat <- 0*H
  corm[[ex]] <- cmat <- 0*H
  if(det(H)!=0) {
#    errm[[ex]] <- emat <- solve(H, tol=1e-24) # covariance matrix
    errm[[ex]] <- emat <- chol2inv(chol(H, pivot=TRUE)) # covariance matrix
    if(any(diag(emat) < 0)) warning("Covariance matrix has negative elements on diagonal.")
    pars$errors <- er <- sqrt(abs(diag(emat))) # parameter errors
    corm[[ex]] <- cmat <- cov2cor(emat)
  }
  # bootstrap estimate of parameter uncertainties
  # number of synthetic data sets (18 -> 17% error on sigma, 50 -> 10%)
  nboot <- 18
  bootm <- matrix(nrow=npar, ncol=nboot) # save parameters for each experiment
  bootm[, 1] <- pa
```

```
  if(BOOTSTRAP){
    if(DEBUG) message("Bootstrapping...")
    npt <- length(x)
    for(jj in 2:nboot){
      # resample data points
      kk <- sample(1:npt, npt, replace=TRUE)
      x <- tfit[[ex]][kk]
      y <- yfit[[ex]][kk]
      # fit
      bfit <- optim(par=ipar, fn=fom, method="L-BFGS-B", lower=lop, upper=upp)
      bootm[, jj] <- bfit$par }
    # define parameters and errors
    pars$fitted <- pa <- apply(bootm, 1, median, na.rm=TRUE)
    pars$errors <- er <- apply(bootm, 1, mad, na.rm=TRUE)
    # recalculate chi2, error matrix, and correlation matrix
    chi2.properties <- fdHess(pa, chisqr)
    chi2[ex] <- chisqr(pa)
    hess[[ex]] <- H <- chi2.properties$Hessian
    errm[[ex]] <- emat <- 0*H
    corm[[ex]] <- cmat <- 0*H
    if(det(H)!=0) {
#     errm[[ex]] <- emat <- solve(H, tol=1e-24) # covariance matrix
      errm[[ex]] <- emat <- chol2inv(chol(H, pivot=TRUE)) # covariance matrix
      if(any(diag(emat) < 0)){
        warning("Covariance matrix has negative elements on diagonal.") }
      corm[[ex]] <- cmat <- cov2cor(emat)
    }
    x <- tfit[[ex]]
    y <- yfit[[ex]] }
  pboot[[ex]] <- bootm
  if(DEBUG){
    message("\nParameters:\n")
    write.table(round(pars, 6), file=stderr(), sep="\t")
    message("\nHessian matrix:\n")
    write.matrix(H, file=stderr())
    message("\nCovariance matrix:\n")
    write.matrix(emat, file=stderr())
    message("\nCorrelation matrix:\n")
    write.matrix(round(cmat, 4), file=stderr())
  }
```

```
  if(DEBUG) message("Chi2 / dof = ", chi2[ex], " / ", dof[ex],
                    " = ", chi2[ex]/dof[ex])
  conv[ex] <- FALSE
  if(fit[[ex]]$convergence==0) conv[ex] <- TRUE
  if(DEBUG) message("Convergence code = ", fit[[ex]]$convergence)
  # save parameters
  lD <- pa[1]
  elD <- er[1]
  D[ex] <- 10^(lD)
  eD[ex] <- 0.5*abs(10^(lD+elD)-10^(lD-elD))
  lDA <- pa[2]
  elDA <- er[2]
  DAi[ex] <- 10^(lDA)
  eDAi[ex] <- 0.5*abs(10^(lDA+elDA)-10^(lDA-elDA))
  lDA <- pa[3]
  elDA <- er[3]
  DAf[ex] <- 10^(lDA)
  eDAf[ex] <- 0.5*abs(10^(lDA+elDA)-10^(lDA-elDA))
#  t0f[ex] <- pa[4]
#  et0f[ex] <- er[4]
#  dtf[ex] <- pa[5]/1e3
#  edtf[ex] <- er[5]/1e3
  FittedLossRate <- function(x) {
    LR <- 0*x
    LR <- loss.rate(t=x, K=Kg, B=Bg[ex], t0=t0f[ex],
                    Ai=DAi[ex]/D[ex], Af=DAf[ex]/D[ex],
                    Jci=Steps$Jci[ex], Jcf=Steps$Jcf[ex],
                    D=D[ex], dt=dtf[ex])
    return(LR)}
  rfit[[ex]] <- y - FittedLossRate(x) # residuals
  if(DEBUG){
    cfit <<- fit
    xxm <- min(x); xxM <- max(x)
    if(xxm<0 & xxM>0){
      xx1 <- seq(xxm, xxm/10, length=5)
      xx2 <- seq(xxm/10, xxM/10, length=101)
      xx3 <- seq(xxM/10, xxM, length=41)
      xx <- c(xx1, xx2, xx3)} else {
      xx <- seq(xxm, xxM, length=41) }
#    yy <- predict(fit, newdata=list(x=xx))
```

```
    yy <- FittedLossRate(xx)
    dev.new(); plot(x, y)
    points(xx, yy, type="l", col="red")
    message("\nFit results:")
    message("\tD = ", D[ex], " +/- ", eD[ex])
    message("\tDAi = ", DAi[ex], " +/- ", eDAi[ex])
    message("\tDAf = ", DAf[ex], " +/- ", eDAf[ex])
#    message("\tt0 = ", t0f[ex], " +/- ", et0f[ex])
#    message("\tdt = ", dtf[ex], " +/- ", edtf[ex])
  }
} # end of loop over experiments
# assemble and save data frame with fit results
OutputVars <- c(OutputVars,
  "explist", "Kg", "Ksmi", "Ksmf", "Bg",
  "t0bef", "t0aft", "ttra",
  "Lbef", "Laft", "Lrmsbef", "Lrmsaft",
  "Dg.pk", "Dg.tr", "eDg.tr", "Dg", "DAig", "DAfg",
  "wwfit", "yyfit", "mmtr", "tfit", "yfit", "wfit", "rfit",
  "dof", "chi2", "fit", "conv", "hess", "errm", "corm", "pboot",
  "D", "eD", "DAi", "eDAi", "DAf", "eDAf", "dtf", "edtf", "t0f", "et0f")
message("Saving results...")
message("List of variables: ", paste(OutputVars, collapse=" "))
save(list=OutputVars, file="tmp_Results.RData")
}


message("Calculating beam distributions...")
# derived quantities
Np <- numeric(nsteps) # number of circulating protons
Nlost.I <- Nlost.L <- numeric(nsteps) # lost protons from intensity and BLM
Ndotp <- numeric(nsteps) # proton loss rate [p/s]
lambdap <- numeric(nsteps)
Flui <- Fluf <- numeric(nsteps) # particle fluxes [p/s]
Ai <- Af <- numeric(nsteps) # beam distribution slopes [p/um^2]
RhoI <- numeric(nsteps) # beam distribution from intensity loss [p/um]
RhoL <- numeric(nsteps) # beam distribution from integrated losses [p/um]
RhoD <- array(data=0, dim=c(nsteps, nsteps)) # and from diffusion model [p/um]

# number of particles, loss rate, and lost particles
dt.lost <- 4 # integration time for calculation of lost particles [s]
```

```
for(ex in explist){
  # number of particles
  if(Steps$plane[ex]=="B1V") fsm <- Nsm.B1 else fsm <- Nsm.B2
  Np[ex] <- predict(fsm, x=Steps$time[ex])$y
  # lost particles
  Nlost.I[ex] <- predict(fsm, x=Steps$time[ex]-dt.lost/2)$y-
                 predict(fsm, x=Steps$time[ex]+dt.lost/2)$y
  tcut <- (Steps$time[ex]-dt.lost/2)<=t.LO & t.LO<=(Steps$time[ex]+dt.lost/2)
  if(Steps$plane[ex]=="B1V")
    lo <- LO.B1V.TOT[tcut]-Bg.B1V else lo <- LO.B2H.TOT[tcut]-Bg.B2H
  Nlost.L[ex] <- mean(lo) / Ksmf[ex]
  if(Steps$plane[ex]=="B1V") fsm <- dNm.B1 else fsm <- dNm.B2
  closest <- which.min(abs(tm-Steps$time[ex]))
  Ndotp[ex] <- -fsm[closest] / dt.cal
}
# decay rate
# particle fluxes
Flui <- DAi / Ksmi
Fluf <- DAf / Ksmf
# distribution slopes
Ai <- Flui / D
Af <- Fluf / D
# density from intensity loss
RhoI <- Nlost.I / Steps$deltaJ
# density from integrated losses
RhoL <- Nlost.L / Steps$deltaJ
# density from slopes and diffusion coefficients
for(exi in 1:nsteps){
  sub0 <- Steps$plane == Steps$plane[exi] &
          Steps$collisions == Steps$collisions[exi] &
          Steps$inw == Steps$inw[exi] &
          D>0
#& 0<(eD/D) & (eD/D)<1
  subi <- sub0 & Steps$Jci <= Steps$Jci[exi]
  iJmax <- which.max(Steps$Jci[sub0])
#  norm <- 0.5*(DAi/Ksmi + DAf/Ksmf)[sub0][iJmax]
  norm <- as.numeric(quantile(0.5*(DAi/Ksmi+DAf/Ksmf)[sub0], probs=0.01))
#  norm <- min(((DAi+DAf)/(Ksmi+Ksmf))[sub0])
  for(exj in 1:nsteps){
    subj <- subi & (Steps$Jci[exj] <= Steps$Jci)
```

```
    RhoD[exi, exj] <- norm * sum((Steps$deltaJ/D)[subj])
#    RhoD[exi, exj] <- sum((0.5*(Flui+Fluf)*Steps$deltaJ/D)[subj])
}}
# comparison of beam tail populations
cases <- list(Steps$plane=="B1V" & !Steps$collisions & Steps$inw,
              Steps$plane=="B1V" & !Steps$collisions & !Steps$inw,
              Steps$plane=="B1V" & Steps$collisions & Steps$inw,
              Steps$plane=="B1V" & Steps$collisions & !Steps$inw,
              Steps$plane=="B2H" & !Steps$collisions & Steps$inw,
              Steps$plane=="B2H" & !Steps$collisions & !Steps$inw,
              Steps$plane=="B2H" & Steps$collisions & Steps$inw,
              Steps$plane=="B2H" & Steps$collisions & !Steps$inw )
nca <- length(cases)
# calculated from beam distribution [protons]
Ntail.f <- list(); length(Ntail.f) <- nca
Ntail.I <- numeric(nca) # calculated from intensity loss [protons]

for(i in 1:nca){
  su <- cases[[i]]
  j <- which(su==TRUE); j1 <- j[1]
  tmin <- min(Steps$tstart[su])
  tmax <- max(Steps$tstop[su])
  if(Steps$plane[j1]=="B1V") fsm <- Nsm.B1 else fsm <- Nsm.B2
  Ntail.I[i] <- predict(fsm, x=tmin)$y - predict(fsm, x=tmax)$y
  for(jj in 1:length(j)){
    if(jj==1) dummy <- numeric(length(j))
    dummy[jj] <- sum(RhoD[j[jj], su]*Steps$deltaJ[su])}
  Ntail.f[[i]] <- dummy
}



message("Calculating beam jitter FFTs...")
sigma.thr <- 3.75
Jitter.cases <- list(
  Steps$plane=="B1V" & !Steps$collisions & Steps$xs<=sigma.thr,
  Steps$plane=="B1V" & Steps$collisions & Steps$xs<=sigma.thr,
  Steps$plane=="B2H" & !Steps$collisions & Steps$xs<=sigma.thr,
  Steps$plane=="B2H" & Steps$collisions & Steps$xs<=sigma.thr)
nca <- length(Jitter.cases)
```

```
Jitter.freq <- Jitter.FFT <- Jitter.FFT.nm <- list()
length(Jitter.freq) <- length(Jitter.FFT) <- length(Jitter.FFT.nm) <- nca
dt <- median(diff(as.numeric(t.LO)))
fmax <- 1/dt
for(ca in 1:nca){
  su <- Jitter.cases[[ca]]
  j <- which(su==TRUE); j1 <- j[1]
  t1 <- min(Steps$tstart[su]); t2 <- max(Steps$tstop[su])
  cut <- t1 <= t.LO & t.LO <= t2
  x <- as.numeric(t.LO[cut]); x <- x-x[1]
  if(Steps$plane[j1]=="B1V") y <- LO.B1V.TOT else y <- LO.B2H.TOT
  y <- y[cut]; y[y<0] <- median(y) # get rid of rare crazy negative values
  n <- length(x)
  Jitter.freq[[ca]] <- seq(0, fmax, length=n)
  w <- get.slepians(n, 1, 3)[,1] # Slepian window
  Jitter.FFT[[ca]] <- Mod(fft(y*w))/n
  # rough calibration
  dy <- diff(range(y))
  dx <- 5e3 # smallest step [nm]
  Jitter.FFT.nm[[ca]] <- Jitter.FFT[[ca]] * dx / dy
}


message("Defining graphics parameter for plotting results...")
Display <- list( col=character(nsteps), pch=numeric(nsteps) )
Display$col[Steps$plane=="B1V"] <- "blue"
Display$col[Steps$plane=="B2H"] <- "red"
Display$pch[Steps$collisions & Steps$inw] <- 16
Display$pch[!Steps$collisions & Steps$inw] <- 1
Display$pch[Steps$collisions & !Steps$inw] <- 15
Display$pch[!Steps$collisions & !Steps$inw] <- 0


getOption("SweaveHooks")[["fig"]]()
message("Plotting diffusion vs. amplitude...")
par(mar=rep(0,4), oma=c(4.2, 5.2, 0.5, 0.5), mgp=c(2,0.5,0), tcl=-0.25)

good <- 0<(eD/D) & (eD/D)<1
#good <- TRUE
# bad fits or outward steps with no recovery
```

```
# good[c(65:68, 121, 146, 195:196, 221, 235:239)] <- FALSE
J <- 0.5*(Steps$Jci+Steps$Jcf)
xl <- range(J[good])
yl <- range(D[good])

sigma.labels <- 0:20
J.at.sigma.labels.B1 <- (sigma.B1.v*sigma.labels)^2/2/
    Lattice.Table["TCP.B1.v", "beta.y"]
J.at.sigma.labels.B2 <- (sigma.B2.h*sigma.labels)^2/2/
    Lattice.Table["TCP.B2.h", "beta.x"]

layout(matrix(1:2, nrow=2, ncol=1))
# beam 1 vertical
cut <- good & explist & Steps$plane=="B1V"
x <- J[cut]
y <- D[cut]
ey <- eD[cut]
co <- Display$col[cut]
sy <- Display$pch[cut]
plot(x, y, xlim=xl, ylim=yl, log="y", type="n", axes=FALSE)
curve(Dp["B1.v"]*x, 0, 2*xl[2], add=TRUE, col="gray") # core emittance growth
points(x, y, col=co, pch=sy)
arrows(x, y-ey, x, y+ey, le=0.05, angle=90, code=3, col=co)
box()
axis(2, las=1)
legend("topleft", bty="n", text.col="blue", legend="B1V")
legend("topright", bty="n", pch=c(1, 0, 16, 15),
       legend=c("Separated beams, inward step",
       "Separated beams, outward step",
       "Colliding beams, inward step",
       "Colliding beams, outward step") )
# beam 2 horizontal
cut <- good & explist & Steps$plane=="B2H"
x <- J[cut]
y <- D[cut]
ey <- eD[cut]
co <- Display$col[cut]
sy <- Display$pch[cut]
plot(x, y, xlim=xl, ylim=yl, log="y", yaxt="n", type="n")
curve(Dp["B2.h"]*x, 0, 2*xl[2], add=TRUE, col="gray")
```

```
points(x, y, col=co, pch=sy)
arrows(x, y-ey, x, y+ey, le=0.05, angle=90, code=3, col=co)
box()
axis(2, las=1)
legend("topleft", bty="n", text.col="red", legend="B2H")
layout(1)
mtext(expression(paste("Action, ", J, " [", mu*m, "]", sep="")),
      1, 3, outer=TRUE)
mtext(expression(paste("Diffusion rate, ", D, " [", mu*m^2/s, "]", sep="")),
      2, 4, outer=TRUE)
mtext(Rubber.Stamp, 1, 3, adj=0, font=3, cex=0.8, outer=TRUE)


getOption("SweaveHooks")[["fig"]]()
message("Plotting beam populations...")
par(mar=rep(0,4), oma=c(4.2, 5.2, 2.2, 3.2), mgp=c(2,0.5,0), tcl=-0.25)

cases <- list(Steps$plane=="B1V" & !Steps$collisions & Steps$inw,
#              Steps$plane=="B1V" & !Steps$collisions & !Steps$inw,
              Steps$plane=="B1V" & Steps$collisions & Steps$inw,
#              Steps$plane=="B1V" & Steps$collisions & !Steps$inw,
              Steps$plane=="B2H" & !Steps$collisions & Steps$inw,
#              Steps$plane=="B2H" & !Steps$collisions & !Steps$inw,
              Steps$plane=="B2H" & Steps$collisions & Steps$inw)
#              Steps$plane=="B2H" & Steps$collisions & !Steps$inw )
nca <- length(cases)
layout(matrix(1:nca, ncol=2, nrow=nca/2, byrow=FALSE))

x <- 0.5*(Steps$Jci+Steps$Jcf)
#xl <- range(x)
xl <- c(0, max(x))
#ymax <- max(c(RhoD[RhoD>0], RhoI[RhoI>0], RhoL[RhoL>0]))
#yl <- c(ymax/1e6, ymax)
yl <- as.numeric(quantile(c(RhoD[RhoD>0], RhoI[RhoI>0], RhoL[RhoL>0]),
                          probs=c(.02, 1)))

for(i in 1:nca){
  su <- cases[[i]]
  j <- which(su==TRUE); j1 <- j[1]
  co <- matlab.colors(length(j))
```

```
  plot(xl, yl, type="n", log="y", axes=FALSE)
  grid(lty="solid")
  if(i==1){ axis(2, las=1); axis(3) }
  if(i==nca){ axis(1); axis(4, las=1) }
  legend.text <- paste(Steps$plane[j1],
                       ifelse(Steps$collisions[j1], "colliding", "separated"),
                       ifelse(Steps$inw[j1], "inward", "outward"))
  legend("topright", bty="n", legend=legend.text,
         text.col=ifelse(Steps$plane[j1]=="B1V", "blue", "red"))
#  if(Steps$inw[j1]){
  if(TRUE){
    t1 <- min(Steps$tstart[j]); t2 <- t1+60
    if(Steps$plane[j1]=="B1V"){
      fsm <- Nsm.B1
      temi <- temit.sl.B1.v
      emi <- emit.sl.B1.v
    } else {
      fsm <- Nsm.B2
      temi <- temit.sl.B2.h
      emi <- emit.sl.B2.h
    }
    N <- predict(fsm, x=t1)$y # initial number of particles
    cut <- t1 <= temi & temi <= t2
    W <- mean(emi[cut])/betar/gammar # initial emittance
    # hypothetical Gaussian distribution
    curve(N*exp(-x/W)/2/W, xl[1], xl[2], col="gray", add=TRUE)
  }
  for(jj in j){
    rug(x[jj], col=co[jj-j[1]+1])
#    Jcmin <- ifelse(Steps$inw[jj], Steps$Jcf[jj], Steps$Jci[jj])
#    Jcmax <- ifelse(Steps$inw[jj], Steps$Jci[jj], Steps$Jcf[jj])
#    slope <- ifelse(Steps$inw[jj], Ai[jj], Af[jj])
#    curve(slope*(Jcmax-x), Jcmin, Jcmax, add=TRUE, col=co[jj-j[1]+1], n=8)
    points(x[su], RhoD[jj, su], col=co[jj-j[1]+1], type="l")
  }
  points(x[su], RhoL[su], col="black", type="l", lty="dashed") # from losses
  points(x[su], RhoI[su], col="black", type="l") # from intensity
  box()
}
```

```
layout(1)
mtext(expression(paste("Action, ", J, " [", mu*m, "]", sep="")),
      1, 3, outer=TRUE)
mtext(expression(
        paste("Beam density, ", f(J), " [protons/", mu*m, "]", sep="")),
      2, 4, outer=TRUE)
mtext(Rubber.Stamp, 1, 3, adj=0, font=3, cex=0.8, outer=TRUE)



getOption("SweaveHooks")[["fig"]]()
message("Plotting beam jitter FFTs...")
par(mar=rep(0,4), oma=c(4.2, 5.2, 2.2, 3.2), mgp=c(2,0.5,0), tcl=-0.25)

cases <- Jitter.cases
nca <- length(cases)
layout(matrix(1:nca, ncol=2, nrow=nca/2, byrow=FALSE))

smo <- 5 # some minimal smoothing
xl <- c(0, fmax/2)
yl <- median(Jitter.FFT.nm[[1]])
for(i in 1:nca) yl <- c(yl, runmed(Jitter.FFT.nm[[i]], smo))
yl <- range(yl)

for(i in 1:nca){
  su <- cases[[i]]; j <- which(su==TRUE); j1 <- j[1]
  x <- Jitter.freq[[i]]
  y <- runmed(Jitter.FFT.nm[[i]], smo)
  plot(x, y, xlim=xl, ylim=yl, type="n", log="y", axes=FALSE)
  grid(lty="solid")
  points(x, y, type="s")
  if(i==1){ axis(2, las=1); axis(3) }
  if(i==nca){ axis(1); axis(4, las=1) }
  legend.text <- paste(Steps$plane[j1],
                       ifelse(Steps$collisions[j1], "colliding", "separated"))
  legend("topright", bty="n", legend=legend.text,
         text.col=ifelse(Steps$plane[j1]=="B1V", "blue", "red"))
  box()
}
layout(1)
mtext("Frequency [Hz]", 1, 3, outer=TRUE)
```

```
mtext("Amplitude [nm, approx.]", 2, 4, outer=TRUE)
mtext(Rubber.Stamp, 1, 3, adj=0, font=3, cex=0.8, outer=TRUE)


message("Making plots of individual steps...")
wi<-6; he<-8; pt<-10; fo<-"Times"
ps.options(width=wi, height=he, pointsize=pt, family=fo, bg="white")
pdf.options(width=wi, height=he, pointsize=pt, family=fo, bg="white")

nana <- length(explist) # total number of analyzed steps
ppp <- 8 # plots per page
pps <- 2 # plots per step: losses vs. time, losses vs. 1/sqrt(t)
# nc <- ceiling(sqrt(3*ppp/4)) # number of columns
nc <- 2
# nr <- ceiling(sqrt(ppp)) # number of rows
nr <- 4
npages <- ceiling(nana * pps / ppp) # number of pages
iex1 <- seq(1, nana, by=ppp/pps) # first and last step indices for each page
iex2 <- pmin(iex1+ppp/pps-1, nana)
ex1 <- explist[iex1] # first and last step numbers for each page
ex2 <- explist[iex2]
for( pn in 1:npages) {
  rangetxt <- paste(formatC(ex1[pn], width=3, flag="0"), "-",
                    formatC(ex2[pn], width=3, flag="0"), sep="")
  basename <- paste("tmp_steps_", rangetxt, sep="")
  pdf(file=paste(basename, ".pdf", sep=""))
  dev.control(displaylist="enable") # enable recording
  par(oma=rep(0,4), mar=c(3.4, 3.4, 0.4, 3.4), mgp=c(2,0.5,0), tcl=-0.25)
  layout(matrix(1:ppp, nrow=nr, ncol=nc, byrow=TRUE))
  for(iex in iex1[pn]:iex2[pn]){
      ex <- explist[iex]
      t <- tfit[[ex]]
      y <- yfit[[ex]]
      t <- t[y>0]
      y <- y[y>0]
      # losses vs. 1/sqrt(t)
      plot(1/sqrt(t[t>0.2]), y[t>0.2])
      legend("topleft", bty="n", cex=0.8,
        legend=paste("Dg.tr = ", formatC(Dg.tr[ex], di=3, format="e"),
                     "+/-", formatC(eDg.tr[ex], di=2, format="e")) )
```

```
# full fit
FittedLossRate <- function(x) {
  LR <- 0*x
  LR <- loss.rate(t=x, K=Kg, B=Bg[ex], t0=t0f[ex],
            Ai=DAi[ex]/D[ex],
            Af=DAf[ex]/D[ex],
            Jci=Steps$Jci[ex], Jcf=Steps$Jcf[ex],
            D=D[ex], dt=dtf[ex])
  return(LR)}
t.peak <- seq(-0.2, 2, by=0.04) # curve near peak
y.peak <- FittedLossRate(t.peak)
yl <- range(c(y, y.peak)) # set limits with space on top
dyl <- diff(yl)
yl[2] <- yl[2]+dyl/2
plot(t, y, col=rgb(0, 0, 0, 0.2), pch=1, cex=0.5, xlab="", ylab="",
     ylim=yl) # data
tm <- c(t.peak, seq(min(t), max(t), length=64)) # model
tm <- sort(tm)
ym <- FittedLossRate(tm)
points(tm, ym, type="l", col="magenta")
legend("topleft", bty="n", cex=0.8,
  legend = c(
    paste("Step #", ex, sep=""),
    paste("time = ", Steps$time.local[ex], sep=""),
    paste("pos. = ", formatC(Steps$cpi[ex], di=3), " to ",
          formatC(Steps$cpf[ex], di=3), " mm", sep=""),
    paste("gap = ",
          formatC(0.5*(Steps$hgi[ex]+Steps$hgf[ex]), di=3),
          " mm", sep=""),
    paste("(", formatC(abs(Steps$xs[ex]), di=3), " sigma)", sep=""),
    paste("J = ", formatC(0.5*(Steps$Jci[ex]+Steps$Jcf[ex]), di=3),
          " um", sep="") ))
legend("topright", bty="n", cex=0.8,
       legend=c(
         paste("D =", formatC(D[ex], di=3, format="e"),
               "+/-", formatC(eD[ex], di=2, format="e")),
         paste("kDAi =", formatC(DAi[ex], di=4),
               "+/-", formatC(eDAi[ex], di=2)),
         paste("kDAf =", formatC(DAf[ex], di=4),
               "+/-", formatC(eDAf[ex], di=2)),
```

```
#               paste("t0 =", formatC(t0f[ex], di=3),
#                     "+/-", formatC(et0f[ex], di=2)),
#               paste("dt =", formatC(dtf[ex], di=3),
#                     "+/-", formatC(edtf[ex], di=2)),
               paste("Chi2 / dof =", formatC(chi2[ex], di=3), "/",
                     dof[ex], "=",
                     formatC(chi2[ex]/dof[ex], di=3))) )
#               paste("conv =", conv[ex])) )
  }
  layout(1)
  mtext("Time [s]", side=1, line=2, adj=0.8)
  mtext("Loss rate [Gy/s]", side=4, line=2)
  dev.copy2eps(file=paste(basename, ".eps", sep=""))
  dev.off()
  IncludePlotInLatex(file=basename, flo=FALSE,
                   cap=paste("Collimator steps ", rangetxt, ".", sep=""),
                   lab=paste("fig:csteps", rangetxt, sep="_"))
}


message("Making steps table...")
# table of steps
Table <- data.frame(time=as.numeric(Steps$time), Steps$time.local, Steps$size,
                    Steps$hgi, ifelse(Steps$collisions, "T", "F"), Steps$plane)
colnames(Table) <- c("\\multicolumn{1}{c}{Unix time~[s]}",
                     "Local time",
                     "\\multicolumn{1}{c}{Size~[$\\mu$m]}",
                     "\\multicolumn{1}{c}{Init. half gap~[mm]}",
                     "Collisions?", "Beam/plane")
rownames(Table) <- 1:nrow(Table)
T <- xtable(Table,
            caption="Table of collimator steps (negative size means inward).",
            label="tab:steps",
            align="rrc..cc",
            digits=c(3, 3, NA, 3, 3, NA, NA),
            display=c("d", "f", "s", rep("fg", 2), rep("s", 2)) )
print(T,
#      floating.environment="table*", floating=TRUE,
      tabular.environment="longtable", floating=FALSE,
      latex.environments=c("ruledtabular"), size="footnotesize",
```

```
      hline.after=0, sanitize.text.function=idem)


message("Making results table...")
Table <- data.frame(Steps$time.local,
                    0.5*(Steps$Jci+Steps$Jcf),
                    Steps$xs,
                    D, eD,
                    DAi, eDAi,
                    DAf, eDAf,
                    chi2, dof)
colnames(Table) <- c("Local time",
                     "\\multicolumn{1}{c}{Action~[$\\mu$m]}",
                     "\\multicolumn{1}{c}{Half gap~[$\\sigma$]}",
                     "$D~\\mathrm{[\\mu m^2/s]}$",
                     "$\\delta D$",
                     "$k D A_i$~[Gy/s]",
                     "$\\delta k D A_i$",
                     "$k D A_f$",
                     "$\\delta k D A_f$",
                     "$\\chi^2$",
                     "dof")
rownames(Table) <- 1:nrow(Table)
T <- xtable(Table,
            caption="Fit results for each step.",
            label="tab:fit.results",
            align="rc..rrrrrrrr",
            digits=c(3, NA, rep(3, 2), 3, 2, 2, 1, 2, 1, 2, 5),
            display=c("d", "s", rep("fg", 2), rep("g", 7), "d") )
print(T,
#      floating.environment="table*", floating=TRUE,
      tabular.environment="longtable", floating=FALSE,
      latex.environments=c("ruledtabular"), size="footnotesize",
      hline.after=0, sanitize.text.function=idem)


message("Making R file for inclusion in LaTeX document...")
Stangle("LHCdiffMD.Rnw", annotate=FALSE) # make R file for inclusion
```

```
message("Cleaning up...")
for(i in 1:3) gc()
message("\n\n", cornicina, "\n",
  "  End of diffusion analysis.\n",
  cornicina, "\n", date(), "\n\n")
```

[1] A. J. Lichtenberg and M. A. Lieberman, *Regular and Chaotic Dynamics* (Springer-Verlag, New York, 1992), p. 320.

[2] T. Chen et al., Phys. Rev. Lett. **68**, 33 (1992).

[3] A. Gerasimov, Report No. FERMILAB-PUB-92-185 (1992).

[4] F. Zimmermann, Part. Accel. **49**, 67 (1995); Report No. SLAC-PUB-6634 (October 1994).

[5] T. Sen and J. A. Ellison, Phys. Rev. Lett. **77**, 1051 (1996).

[6] N. Mokhov et al., JINST **6**, T08005 (2011).

[7] L. Burnod, G. Ferioli, and J. B. Jeanneret, Report No. CERN-SL-90-01 (1990).

[8] R. P. Fliller III et al., in *Proc. 2003 Part. Accel. Conf. (PAC03)*, p. 2904 (IEEE, Piscataway, NJ, 2003).

[9] G. Stancari et al., Phys. Rev. Lett. **107**, 084802 (2011), arXiv:1105.3256 [physics.acc-ph].

[10] G. Stancari, Report No. FERMILAB-FN-0926-APC, arXiv:1108.5010 [physics.acc-ph].

[11] G. Stancari et al., Measurements of transverse beam diffusion rates in the Tevatron collider, Proc. 2nd Int. Part. Accel. Conf. (IPAC11), San Sebastián, Spain, 4–9 September 2011, p. 1882.

[12] K.-H. Mess and M. Seidel, Nucl. Instr. Meth. Phys. Res. A **351**, 279 (1994); M. Seidel, PhD Thesis, Hamburg University, DESY 94-103 (June 1994).

[13] R Development Core Team, *R: A language and environment for statistical computing* (R Foundation for Statistical Computing, Vienna, Austria, 2010), ISBN 3-900051-07-0, www.R-project.org.

[14] Friedrich Leisch, Sweave: Dynamic generation of statistical reports using literate data analysis, in Compstat 2002 — Proceedings in Computational Statistics (Wolfgang Härdle and Bernd Rönz, editors), pp. 575–580 (Physica Verlag, Heidelberg, 2002), ISBN 3-7908-1517-9.

[15] G. Valentino et al., Halo diffusion MD proposal (2012, unpublished).

[16] G. Valentino et al., Halo scraping, diffusion and repopulation MD, CERN-ATS-Note-2012-XXX-MD (2012).