Supplementary Information

Non-insecticidal vineyard pesticides adversely affect the near-extinct Moselle Apollo butterfly

Jan Erik Sedlmeier[1*], Emma Gamer[1], Lorenz Witt[2], Claus Denzlinger[1], Sarah Rißmann[1], Denise Kuhn[1], Daniel Müller[3], Tim Laußmann[3], Carsten A Brühl[2*], Georg Petschenka[1,4*]

[1]Department of Applied Entomology, University of Hohenheim, Stuttgart, Germany
[2]Community Ecology & Ecotoxicology, Technical University of Kaiserslautern-Landau, Landau, Germany
[3]Melanargia - Arbeitsgemeinschaft Rheinisch-Westfälischer Lepidopterologen e.V., Leverkusen, Germany
[4]KomBioTa – Center for Biodiversity and Integrative Taxonomy, University of Hohenheim and State Museum of Natural History, Stuttgart, Germany

*Corresponding authors:
Jan Erik Sedlmeier -> jan.sedlmeier@uni-hohenheim.de
Georg Petschenka -> georg.petschenka@uni-hohenheim.de
Carsten A Brühl -> carsten.bruehl@rptu.de

This document includes:

Supplementary Methods S1 – General Housing and Environmental Conditions 3

Supplementary Methods S2 – Misting of Larval Housing Cups and Watering of *Sedum album* Plants Sprayed with Fungicide 3

Supplementary Table S1 – GPS data for each plant sample collected in the Moselle Valley at the four locations between February 29th and March 1st 2024. 3

Supplementary Methods S3 – Detailed Pesticide Residue Analysis in the Field-Collected Host Plant Experiment 5

Supplementary Tables S2 to S6 – Pesticide analytical data of *Sedum album* samples from the Field-Collected Host Plant Experiment. 9

Supplementary Methods S4 – Detailed Statistical Analysis of the Field-Collected Host Plant Experiment 13

Supplementary Methods S5 – Helicopter Spray Experiments: Spray Chamber Configuration, Application Calculations and Additional Information 15

Supplementary Tables S7 to S10 16

Supplementary Methods S6 – Detailed Methodology of the Luna® Experience Toxicity Experiments 17

Supplementary Tables S11 to S15 18

Supplementary Figure S1 – Adult Sex Ratios in the Luna® Experience Toxicity Experiments 21

Supplementary Figure S2 – Wing deformation in the Luna® Experience Toxicity Experiment 22

Supplementary Figure S3 – Growth of *P. a. vinningensis* larvae in individual housing during the Luna® Experience toxicity experiment. 23

Supplementary Methods S7 – Terminated Luna® Experience Toxicity Experiment 23

Supplementary Figure S3: Growth of *Parnassius apollo vinningensis* larvae in individual housing during the excluded Luna® Experience toxicity experiment. ... 24

Supplementary Table S16 – Data collected in the terminated Luna® Experience trial 25

Supplementary Analysis S1 – Determination of *Sedum album* Plant Nutrient Contents from the Field-Collected Host Plant Experiment and Larval Body Mass / Mode-of-Action Diversity Correlation Analysis..... 30

Supplementary Figure S4: Temperature program of the microwave digestion in the plant nutrient analysis. 31

Supplementary Table S17: Results for the Spearman's rank correlation analysis. 32

Supplementary Figure S5: Relationship between plant nutrient content (nitrogen, phosphorus, copper, sulfur, potassium) and mean larval weight per plant after 31 days of exposure..... 33

Supplementary Figure S6: Relationship between plant nitrogen content and mode-of-action diversity. 33

Statistical Report and Creation of Graphs 35

Supplementary Methods S1 – General Housing and Environmental Conditions

Larvae were maintained in ventilated plastic containers covered with insect mesh under greenhouse or climate chamber conditions. Unless stated otherwise, larvae were handled using fine brushes and forceps, and host plants were provided ad libitum. To standardize disturbance among treatments, feeding and maintenance procedures were conducted simultaneously across experimental units: when plant material was either completely consumed (early larval stages: all young shoots removed) or began to dry out, food was added to all cups and dried material removed to ensure equal disturbance, and the amount of food added was kept constant across cups

Greenhouse experiments were conducted on mobile rail-mounted trolleys positioned inside a greenhouse equipped with winter-protection ventilation preventing temperatures below 10 °C. When outdoor temperatures exceeded 10 °C and weather conditions were suitable (no precipitation or strong wind), trolleys were temporarily placed outdoors during daytime and returned indoors before night-time temperatures declined below 10 °C.

Supplementary Methods S2 – Misting of Larval Housing Cups and Watering of *Sedum album* Plants Sprayed with Fungicide

In the Field-Collected Host Plant Experiment, during the first seven days, cups were manually misted twice daily (7:00 & 18:00 h) to delay plant desiccation. An automated misting system equipped with 0.4 mm QUANTIC© duo mist nozzles (Micro Rain Systems e.K., Germany) was subsequently installed above the cups on each trolley to standardize misting and deliver moisture more evenly across the day. The system delivered fine mist four times daily (07:00, 12:00, 16:00, and 20:00 h). The same misting system was used in the Helicopter Spray Experiment starting 48 h after fungicide applications.

S. album plants planted in plastic trays that were sprayed with fungicide and harvested during experiments to feed *P. a. vinningensis* larvae were regularly watered starting 48 h after fungicide application. *S. album* used in the Luna® Experience trial with grouped larvae was completely harvested before trial onset and therefore not watered after fungicide application.

Supplementary Table S1 – GPS data for each plant sample collected in the Moselle Valley at the four locations between February 29th and March 1st 2024.

Plant ID (within location)	Location	Lat	Lon
1	Brauselay	50.14173479	7.18756394
2	Brauselay	50.14190883	7.18784456
3	Brauselay	50.14203845	7.18808229
4	Brauselay	50.14213681	7.18814852

5	Brauselay	50.14218939	7.18780113
6	Brauselay	50.14216264	7.18738429
7	Brauselay	50.14215669	7.18680559
8	Brauselay	50.14232791	7.18713537
9	Brauselay	50.14245042	7.18733307
10	Brauselay	50.14258617	7.18813829
11	Brauselay	50.14276146	7.18927918
12	Brauselay	50.14289754	7.19000116
13	Brauselay	50.1429536	7.1908643
14	Brauselay	50.14306363	7.19166616
15	Brauselay	50.14315036	7.19258709
1	Calmont	50.10416108	7.1376741
2	Calmont	50.104246	7.13719964
3	Calmont	50.10454696	7.13690209
4	Calmont	50.10504366	7.13618857
5.1	Calmont	50.10495283	7.13585617
5.2	Calmont	50.1053038	7.13620545
6	Calmont	50.10477845	7.1362995
7	Calmont	50.10490225	7.13558134
8	Calmont	50.10432074	7.13698003
9	Calmont	50.10445422	7.13887767
10	Calmont	50.10367537	7.13892315
1	Rosenberg	50.31588144	7.46214892
2	Rosenberg	50.31604803	7.46194832
3	Rosenberg	50.31614497	7.46183004
4	Rosenberg	50.3161744	7.46170912
5	Rosenberg	50.31647133	7.46156858
6	Rosenberg	50.3165053	7.4616879
7	Rosenberg	50.3165392	7.4618632
8	Rosenberg	50.31652148	7.46160068
9	Rosenberg	50.31642361	7.46140063
10	Rosenberg	50.3162759	7.46167333
11	Rosenberg	50.31631052	7.46178391
12	Rosenberg	50.31631215	7.46189223
13	Rosenberg	50.31660415	7.46276428
14	Rosenberg	50.31671088	7.46254077
15	Rosenberg	50.31689703	7.46255102
16	Rosenberg	50.31677962	7.4626373
18	Rosenberg	50.31765701	7.46661667
19	Rosenberg	50.31751219	7.46639313
20	Rosenberg	50.31791638	7.46704848
21	Rosenberg	50.31815592	7.46743847
22	Rosenberg	50.31839328	7.46734112
23	Rosenberg	50.31770423	7.465732
24	Rosenberg	50.31770356	7.46568114
25	Rosenberg	50.31774601	7.46569935
26	Rosenberg	50.3175752	7.4653303
27	Rosenberg	50.31757398	7.46521055

28	Rosenberg	50.31762874	7.46516062
29	Rosenberg	50.31766966	7.46498258
30	Rosenberg	50.31778437	7.46539284
31	Rosenberg	50.31774905	7.46546201
32	Rosenberg	50.31778437	7.46539284
33	Rosenberg	50.31795414	7.46522811
1	Dortebach	50.168717	7.213244
2	Dortebach	50.168778	7.213492
3	Dortebach	50.168883	7.213006
4	Dortebach	50.168725	7.212567
5	Dortebach	50.16905	7.212481
6	Dortebach	50.169156	7.212725
7	Dortebach	50.169075	7.212936

Supplementary Methods S3 – Detailed Pesticide Residue Analysis in the Field-Collected Host Plant Experiment

For each *Sedum album* accession, composite plant samples were collected concurrently with experimental setup. Approximately 7 g fresh plant material per accession was transferred into 50 mL centrifuge tubes, stored at −80 °C, and transported on dry ice to the Community Ecology and Ecotoxicology Laboratory at RPTU Kaiserslautern–Landau.

The wet weight of the frozen *Sedum* samples was determined on a fine scale (Mettler Toledo ME802-1, Mettler-Toledo GmbH, Greifensee, Schweiz) and subsequently the samples were freeze dried (Alpha 1-4 LCSbasic, Martin Christ Gefriertrocknungsanlangen GmbH, Osterode am Harz, Germany) for at least 70 hours. Then the samples were milled in a knife mill (Grindomix GM 300, Retsch, Haan, Germany) for 1 min at 3000 rpm to avoid heating and afterwards the dry weight was determined on a fine scale (Mettler Toledo ME802-1, Mettler-Toledo GmbH, Greifensee, Schweiz.

Validated multicomponent analytical methods (Bakanov et al., 2023) were used to extract 93 CUPs (36 fungicides, 36 herbicides, 21 insecticides) from *Sedum*. In brief, 0.50 g of vegetation was extracted with 10 mL of acetonitrile (MeCN, HPLC gradient grade ≥ 99.9%, Honeywell, Charlotte, USA) containing 2.5% formic acid (CH_2O_2, HiPerSolv Chromanorm ≥ 99.0%, VWR, Radnor, USA) and 250 mg of salt (ammonium formate, NH_4HCO_2, reagent grade ≥ 99.0%, Sigma–Aldrich, St. Louis, USA). If the initial volume of the respective *Sedum* sample was ≤ 0.6 mg, the sample volume was taken half so that 0.25 mg of dry *Sedum* and 5 mL of AcN with 2.5 % FA were used. Affected samples are marked with “X2” after the sample name in the respective cells in the supplementary tables S2–S6 below. The samples were placed in an overhead shaker for 1 h (Stuart drive Rotator drive STR4, ColeParmer, Vernon Hills, USA); afterwards, the samples were centrifuged (MegaStar 1.6R, VWR, Radnor, USA, 6 min at 3000 rpm) and filtered (17 mm HPLC syringe filter, PTFE, pore size 0.2 µm; La-Pha-

Pack, Langerwehe, Germany). The vegetation extracts were further cleaned with graphitised carbon black (GCB, Carbon SPE Bulk Sorbent, Agilent Technologies, Santa Clara, USA). For the clean-up, 1 ml of vegetation extract was added to 7.5 mg of GCB in a 2 ml Eppendorf tube. The tubes were vortexed for 60 s and the supernatant was filtered through 0.2 µm filters directly into the LC-MS vials.

The extracts were analyzed by high-performance liquid chromatography-tandem to triple quadrupole mass spectrometry via electrospray ionisation (HPLC)-ESI-MS/MS (HPLC: Agilent Technologies LC 1260 Infinity II series, MS/MS: Agilent Technologies 6495 C, Santa Clara CA, USA). The instrumental performance and potential carry-over effects were controlled by injecting quality control and analytical blank samples, respectively. For quality control, all the samples were spiked with 25 µL of the deuterated standard imidacloprid-D4 (98.9%, Dr. Ehrenstorfer GmbH, Augsburg, Germany) in acetonitrile (MeCN, HPLC gradient grade ≥ 99.9%, Honeywell, Charlotte, USA) (concentration: 10 mg/L). The solvent was allowed to evaporate for 30 min under a fume hood. The peak areas of imidacloprid-D4 in the samples and of the respective blank samples were compared to check for extraction quality.

Positive confirmation of an analyte was done with following criteria: (1) The retention time of the peak had to match the standard sample in a similarly concentrated calibration standard (SANTE, 2024). (2) The qualifier ion ratios had to fall within the range of 70 – 120 %. (3) One S/N ratio had to be higher than 10 and the second S/N ratio had to higher than 3 (Rösch et al., 2023) (4) Pesticides detected below the determined limit of quantification (LOQ) were considered as detected but for concentration-specific questions, only measured residues above the LOQ were used. The limit of quantification (LOQ) of each analyte was determined as the lowest concentration in the calibration series which satisfied the above criteria and also had a signal-area relative standard deviation below 15% (%RSD). The HPLC-MS/MS data was processed using Agilent Mass Hunter Quantitative Analysis software (Version 10.0., Build 10.0.707.0, Agilent Technologies, 2006 – 2018, Santa Clara CA, USA). Calibration curves demonstrated satisfying linearity for the covered concentration range ($R2 \geq 0.99$). A weighting factor of $1/x^2$ was applied to the calibration curve if necessary, as it has been shown to be optimal for bioanalytical data (Almeida et al., 2002), improving curve fit and quantification results for low concentrations (Dolan, 2009).

The number of the CUPs was derived by adding all the substances detected in one *Sedum* sample. The CUP sum concentrations were calculated as the sum of all single CUPs detected above the limit of quantification (>LOQ) for each *Sedum* sample (Supplementary Tables S2–S6 below). For each *S. album* sample, the total number of detected pesticides, summed pesticide concentration, and herbicide/fungicide mode-of-action diversity (HRAC/FRAC classification) were also calculated. Residues below LOQ were included in pesticide counts but excluded from concentration-based analyses (see Supplementary Data).

LC-MS Parameters:

High performance liquid chromatography (HPLC) parameters

Instrument:	Agilent 1260 Infinity II HPLC System	
Column:	ZORBAX Eclipse Plus C18 (3.0 ID x 150 mm, 2.7 micron)	
Eluent A:	H_2O/MeOH (98:2), 0.1% Formic acid, 4 mM Ammonium formate	
Eluent B:	H_2O/MeOH (2:98), 0.1% Formic acid, 4 mM Ammonium formate	
Injection volume:	1 µL	
Flow rate:	0.45 mL/min	
Column temperature:	45°C	
Elution gradient:	Time (min)	% Eluent A
	0	98
	1	50
	4	35
	14	0
	20	0
	20.1	98

Electrospray ionization (ESI) and triple quadrupole mass spectrometry (MS/MS) parameters

Instrument:	Agilent 6495 Triple Quadrupole Mass Spectrometer with an iFunnel Jet Stream ESI
Capillary voltage:	3000 V
Nozzle voltage:	0 V
Gas flow:	11 L/min
Gas temperature:	250 °C
Sheath gas flow:	12 L/min
Sheath gas temperature:	350 °C
Nebulizer pressure:	38 psi

References

Almeida, A. M., Castel-Branco, M. M., & Falcão, A. C. (2002). Linear regression for calibration lines revisited: Weighting schemes for bioanalytical methods. *Journal of Chromatography B*, *774*(2), 215–222. https://doi.org/10.1016/S1570-0232(02)00244-1

Bakanov, N., Honert, C., Eichler, L., Lehmann, G. U. C., Schulz, R., & Brühl, C. A. (2023). A new sample preparation approach for the analysis of 98 current-use pesticides in soil and herbaceous vegetation using HPLC-MS/MS in combination with an acetonitrile-based extraction. *Chemosphere*, *331*, 138840. https://doi.org/10.1016/j.chemosphere.2023.138840

Dolan, J. (2009). *Calibration Curves, Part II: What are the Limits? 27*, 306–312.

Rösch, A., Wettstein, F. E., Wächter, D., Reininger, V., Meuli, R. G., & Bucheli, T. D. (2023). A multi-residue method for trace analysis of pesticides in soils with

special emphasis on rigorous quality control. *Analytical and Bioanalytical Chemistry*, *415*(24), 6009–6025. https://doi.org/10.1007/s00216-023-04872-8

SANTE. (2024). *SANTE 11312/2021 v2: Analytical Quality Control and Method Validation Procedures for Pesticide Residues Analysis in Food and Feed* (Guidance Document SANTE 11312/2021 Rev. 2). European Commission, Directorate-General for Health and Food Safety (DG SANTE).

Supplementary Tables S2 to S6 – Pesticide analytical data of *Sedum album* samples from the Field-Collected Host Plant Experiment. The following tables list the data for the locations Rosenberg (“Ros”, Table S2), Calmont (“Cal”, Table S3), Brauselay (“Braus”, Table S4), Dortebach (“Dorte”, Table S5), the control food and the total number of detections per pesticide across all locations (Table S6). Numbers are given in ng/g. “X2” indicates samples extracted at half scale due to limited sample mass (see Supplementary Methods 3 above). The last two rows of each table indicate the total number of individual pesticides detected per plant sample and the sum concentration of all pesticides detected per sample.

Table S4.2: Rosenberg

Pesticide	Pesticide Class	Ros 1 X2	Ros 2	Ros 3	Ros 4 X2	Ros 5	Ros 6	Ros 7	Ros 8	Ros 9	Ros 11	Ros 12	Ros 13	Ros 14	Ros 15	Ros 16 X2	Ros 17	Ros 18	Ros 19	Ros 21	Ros 22	Ros 23 X2	Ros 26	Ros 28 X2	Ros 30	Ros 31	Ros 33
Boscalid	Fungicide	5.1			2.26	3.51			0.85					3.49		1.99	<LOQ			5.86							
Cyazofamid	Fungicide												0.46														
Cyflufenamid	Fungicide		1.37	3.41	0.82	2.68	2.14	1.44	0.58	0.55		2.16	1.5	1.97		2.52			<LOQ	1.82	4.79	9.73					
Cyprodinil	Fungicide																										
Difenoconazole	Fungicide			9.59			2.78																				
Dimethomorph	Fungicide	1.74	2.15	16.97	1.6	7.76	9.46	3.62	1.91	<LOQ	5.16	4.29	1.73	50.76		5.67	3.75			7.39	7.53	18.6					
Fluopicolide	Fungicide	14.33	6.23	5.32		9.53	6.43	4.4		7.72		9.46	16.81	11.59	21.58	18.17	14.28			14.63	24.67	43.21	<LOQ				
Fluopyram	Fungicide	10.28	15.41	6.6	31.19	19.34	12.96	7.41	3.05	13.04	4.97	17.52	21.07	16.27	20.95	33.47	28.84	1.75	1.23	15.34	9.07	27.03		0.72			
Fluxapyroxad	Fungicide	4.38	3.44	9.9	8.14	8.52	8.86	5.5		2.3	3.74	7.98		24.73	5.04	12.08	11.85		0.64	12	12.98	31.69					
Metrafenone	Fungicide	1.21		4.33		1.56	4.01				0.65		0.88	7.11	1.02	2.39	0.81			2.1	2.28	6.54					
Myclobutanil	Fungicide																										
Penconazole	Fungicide															<LOQ											
Spiroxamine	Fungicide																										
Tebuconazole	Fungicide			<LOQ		<LOQ		<LOQ						<LOQ													
Diflufenican	Herbicide																										
Metamitron	Herbicide																			<LOQ							
Metobromuron	Herbicide																										
Propyzamide	Herbicide															0.26						0.13					
Number		6	5	8	5	8	7	6	4	5	4	5	6	8	4	9	6	1	3	8	6	7	1	1	0	0	0
Sum Con. [ng/g]		37.04	28.6	56.12	44.01	52.9	46.64	22.37	6.39	23.61	14.52	41.41	42.45	115.92	48.59	76.55	59.53	1.75	1.87	59.14	61.32	136.93	0	0.72	0	0	0

Table S3: Calmont

Pesticide	Pesticide Class	Cal 1	Cal 2	Cal 4 X2	Cal 5.1 X2	Cal 5.2	Cal 6	Cal 7	Cal 8	Cal 9	Cal10
Boscalid	Fungicide		1.16		1.19			17.41	10.42		
Cyazofamid	Fungicide										
Cyflufenamid	Fungicide		1.16			0.81			0.57		
Cyprodinil	Fungicide										
Difenoconazole	Fungicide										
Dimethomorph	Fungicide		3.76	1.85	0.97	1.18		1.15			
Fluopicolide	Fungicide		6.08	5.58				4.24	7.4		
Fluopyram	Fungicide	0.64		3.02	0.96	1.13	0.18		6.9		
Fluxapyroxad	Fungicide	<LOQ	3.19			1.05		4.58	13.36		
Metrafenone	Fungicide							0.67			
Myclobutanil	Fungicide								0.58		
Penconazole	Fungicide								1.77		
Spiroxamine	Fungicide								0.22		
Tebuconazole	Fungicide										
Diflufenican	Herbicide										
Metamitron	Herbicide										
Metobromuron	Herbicide										
Propyzamide	Herbicide										
Number		2	5	3	3	4	1	5	8	0	0
Sum Con. [ng/g]		0.64	15.35	10.45	3.12	4.17	0.18	28.05	41.22	0	0

Table S4: Brauselay

Pesticide	Pesticide Class	Braus 1	Braus 2	Braus 3	Braus 4	Braus 5	Braus 6	Braus 7	Braus 8	Braus 9	Braus 10	Braus 11	Braus 12	Braus 13	Braus 14	Braus 15
Boscalid	Fungicide		1.98	3.05	2.4											
Cyazofamid	Fungicide		0.59													
Cyflufenamid	Fungicide	2.23	1.92													
Cyprodinil	Fungicide	1.24	4.93													
Difenoconazole	Fungicide	3.75														
Dimethomorph	Fungicide	3.88														
Fluopicolide	Fungicide	5.21	18.35	6.39	3.9											
Fluopyram	Fungicide	8.42	27.84	15.08	9.63									0.54		
Fluxapyroxad	Fungicide	3.33	3.49	2.45												
Metrafenone	Fungicide	0.75														
Myclobutanil	Fungicide															
Penconazole	Fungicide															
Spiroxamine	Fungicide															
Tebuconazole	Fungicide															
Diflufenican	Herbicide															
Metamitron	Herbicide															
Metobromuron	Herbicide												<LOQ			
Propyzamide	Herbicide															
Number		8	7	4	3	0	0	0	0	0	0	0	1	1	0	0
Sum Con. [ng/g]		28.81	59.1	26.97	15.93	0	0	0	0	0	0	0	0	0.54	0	0

Table S5: Dortebach

Pesticide	Pesticide Class	Dorte 1 X2	Dorte 2 X2	Dorte 3	Dorte 4	Dorte 5	Dorte 6	Dorte 7 X2
Boscalid	Fungicide							
Cyazofamid	Fungicide							
Cyflufenamid	Fungicide							
Cyprodinil	Fungicide							
Difenoconazole	Fungicide							
Dimethomorph	Fungicide							
Fluopicolide	Fungicide							
Fluopyram	Fungicide							
Fluxapyroxad	Fungicide							
Metrafenone	Fungicide							
Myclobutanil	Fungicide							
Penconazole	Fungicide							
Spiroxamine	Fungicide							
Tebuconazole	Fungicide							
Diflufenican	Herbicide	0.1			<LOQ			
Metamitron	Herbicide							
Metobromuron	Herbicide							
Propyzamide	Herbicide							
Number		1	0	0	1	0	0	0
Sum Con. [ng/g]		0.1	0	0	0	0	0	0

Table S6: Control food & total pesticide number across locations

Pesticide	Pesticide Class	Control Food Mix	Total Number of Detections
Boscalid	Fungicide		15
Cyazofamid	Fungicide		2
Cyflufenamid	Fungicide		21
Cyprodinil	Fungicide		2
Difenoconazole	Fungicide		3
Dimethomorph	Fungicide		24
Fluopicolide	Fungicide		25
Fluopyram	Fungicide		33
Fluxapyroxad	Fungicide		26
Metrafenone	Fungicide		15
Myclobutanil	Fungicide		1
Penconazole	Fungicide		2
Spiroxamine	Fungicide		1
Tebuconazole	Fungicide		4
Diflufenican	Herbicide		2
Metamitron	Herbicide		1
Metobromuron	Herbicide		1
Propyzamide	Herbicide		2
Number		0	
Sum Con. [ng/g]		0	

Supplementary Methods S4 – Detailed Statistical Analysis of the Field-Collected Host Plant Experiment

Pesticide Predictors

To test for an effect of environmental pesticides on caterpillar growth and mortality, three pesticide-related predictor types were evaluated: (1) *total pesticide concentration* (ng/g dry weight), (2) *number of pesticides* detected (ranging from 0 to 9), and (3) pesticide MoA groups present. The latter was divided into (a) individual MoA groups present, represented as binary presence/absence variables and (b) MoA indices: (i) *herbicide MoA diversity* (distinct HRAC groups present; 0–1), (ii) *fungicide MoA diversity* (distinct FRAC groups; 0–6) and (iii) *total MoA diversity* (combined number of distinct HRAC/FRAC groups; 0–6). The distribution of these predictors across sampling locations is presented in figure 4.2. This allows to test whether larval growth or survival can be predicted by pesticide concentration or number, the presence of individual MoA groups, or the diversity of MoA groups. To further investigate if effects are linear or threshold-based, some predictors were tested both as continuous and categorical variables. These included *number of pesticides* (continuous: 0–9; categorical: 0, 1–4, 5–9), *fungicide MoA diversity* (continuous: 0–6; categorical: 0, 1–3, 4–6) and *total MoA diversity* (continuous: 0–6; categorical: 0, 1–3, 4–6). Categorisations were based on data distribution to ensure similar sample sizes across categories. *Herbicide MoA diversity* was not categorised as it only ranged from 0 to 1.

As pesticide exposure was not experimentally controlled but quantified afterwards, larval responses were analysed by a systematic model comparison to evaluate whether pesticide predictors added explanatory value, as described in the following sections.

Larval Weight

Two modelling approaches were considered: (1) larval growth across all sampling days (17, 24, and 31 days of exposure), and (2) final larval weight after 31 days, as chronic pesticide effects on weight were expected to be most evident in later stages. Linear mixed-effects models (LMMs) were fitted in lme4 (Bates et al. 2015). Sampling location was included as a covariate to account for site-specific differences unrelated to pesticide contamination. Plant ID (experimental unit) was included as a random intercept to account for non-independence of larvae reared on the same plant sample. For growth analyses, larval weight was log-transformed to meet model assumptions, and time since exposure was centred and included as a continuous predictor with linear and quadratic terms to allow for non-linear trajectories. An initial random-effects structure including random slopes of time within plant ID resulted in singular fits and was therefore simplified to random-intercept models. Corresponding null models for growth and final weight were expanded by adding pesticide predictors, each fitted in separate models to avoid collinearity among exposure metrics. Both numerical and categorical predictors were analysed, and additive and interaction structures were tested. Interaction models included pesticide × location terms and, in growth analyses, pesticide × time interactions (linear and/or quadratic. For individual MoA groups, additive screening models including all detected groups were compared with reduced models representing selected or biologically motivated group combinations.
Model support was evaluated using maximum likelihood–based AIC. Following Burnham and Anderson (2002), models with $\Delta AIC \leq 2$ were considered to have substantial support. All LMMs were fitted using maximum likelihood for model comparison. Model assumptions were assessed using visual residual diagnostics (performance package; Lüdecke et al. 2021). Support for pesticide effects was evaluated based on AIC comparison, effect sizes, and their 95% CIs. Because some pesticides were detected but not quantifiable (<LOQ), models including total pesticide concentration were fitted to a reduced dataset. Since AIC values are only comparable among models fitted to identical response data, these models were evaluated in a separate candidate model set. Prior to analysis, total pesticide concentration was standardised (z-transformed).

Larval Survival

Larval survival was analysed using binomial generalized linear models (GLMs; logit link) implemented in lme4 (Bates et al. 2015). Survival was defined as the proportion of larvae alive per plant ID (housing cup) after 31 days of exposure. As in the weight analyses, a systematic model-screening approach was applied to the same pesticide predictors. The null model included location as a mandatory covariate and was expanded by adding total *number of pesticides*, *fungicide MoA diversity*, and *total MoA*

diversity, each fitted in separate models to avoid collinearity. Predictors were analysed as both numerical and categorical terms. For numerical predictors, pesticide × location interaction models were additionally fitted Individual MoA groups and *herbicide MoA diversity* were not analysed due to highly uneven predictor distributions and limited variation in survival, which would risk unstable parameter estimation in binomial GLMs. *Total pesticide concentration* was again analysed separately; the only converging model included z-transformed concentration as an additive covariate alongside location. Prior to model comparison, fitted probabilities were inspected to exclude separation or extreme predictions. Model support was assessed using AIC. Model assumptions were evaluated using DHARMa-based simulated residual diagnostics (Hartig 2024), including checks for overdispersion. Predictor effects were interpreted based on odds ratios (OR) and 95% confidence intervals.

References

Bates D, Mächler M, Bolker B, Walker S (2015) Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software 67:1–48. https://doi.org/10.18637/jss.v067.i01

Burnham KP, Anderson DR, Huyvaert KP (2011) AIC model selection and multimodel inference in behavioral ecology: some background, observations, and comparisons. Behav Ecol Sociobiol 65:23–35. https://doi.org/10.1007/s00265-010-1029-6

Hartig F (2024) DHARMa: Residual Diagnostics for Hierarchical (Multi-Level / Mixed) Regression Models. R package version 0.4.7, <https://CRAN.R-project.org/package=DHARMa>

Lüdecke D, Ben-Shachar MS, Patil I, et al (2021) performance: An R Package for Assessment, Comparison and Testing of Statistical Models. Journal of Open Source Software 6:3139. https://doi.org/https://doi.org/10.21105/joss.03139

Supplementary Methods S5 – Helicopter Spray Experiments: Spray Chamber Configuration, Application Calculations and Additional Information

To replicate the helicopter fungicide application regime conducted on viticultural slopes between Cochem-Cond and Valwig in 2023 (Table 4.2, main document), documented product combinations and application rates per hectare were used as the basis for all spray calculations. Because spray volumes were not documented, an application volume of 150 L water ha^{-1} was assumed, corresponding to the approved minimum spray volume for aerial fungicide applications in viticulture (BVL 2024).

For each application event, the documented amounts of formulated products and the assumed spray volume were proportionally scaled to the spray chamber area (2 m^2).

To simulate spray drift exposure, fungicide amounts were further reduced to 6.68% of the field application rate based on published helicopter drift deposition estimates for vineyards at 20 m distance (JKI 2020), while maintaining constant spray volume. Detailed calculations for all four application events are provided in the tables below.

References:

BVL (2024) Liste der Pflanzenschutzmittel, die für die Anwendung mit Luftfahrzeugen zugelassen bzw. genehmigt sind. https://doi.org/https://www.bvl.bund.de/SharedDocs/Downloads/04_Pflanzenschutzmittel/psm_hubschrauber.pdf?__blob=publicationFile&v=25

JKI (2020) Tabelle der Abdrifteckwerte für Einfachanwendungen - Stand: 18. Mai 2020. Julius Kühn-Institut (JKI), Quedlinburg, Germany

Supplementary Tables S7 to S10 – Calculation of the simulation of helicopter applications in the spray chamber by application day in the Helicopter Spray Experiment. Each application was conducted as a tank mixture.

Table S7: 1st Application – Day 0

Product	Application per Hectare		Application in Spray Chamber (2 m^2)		
	Amount of Product Used	Assumed Amount of Water Used	Amount of Product - Field Rate (100%)	Amount of Product - Drift Rate (6.68%)	Amount of Water
Delan® Pro	3 L	150 L	0.6 ml	40.08 µL	0.03 L
Netzschwefel (Sulfur)	4.5 kg		0.9 g	60.12 mg	

Table S8: 2nd Application – Day 11

Product	Application per Hectare		Application in Spray Chamber (2 m^2)		
	Amount of Product Used	Assumed Amount of Water Used	Amount of Product - Field Rate (100%)	Amount of Product - Drift Rate (6.68%)	Amount of Water
Delan® Pro	3 L	150 L	0.6 ml	40.08 µL	0.03 L
Netzschwefel (Sulfur)	4.5 kg		0.9 g	60.12 mg	
Vivando®	0.2 L		40 µL	2.67 µL	

Table S9: 3rd Application – Day 21

Product	Helicopter Application per Hectare		Application in Spray Chamber (2 m^2)		
	Amount of Product Used	Assumed Amount of Water Used	Amount of Product - Field Rate (100%)	Amount of Product - Drift Rate (6.68%)	Amount of Water
Profiler®	2.31 kg	150 L	468 mg	31.26 mg	0.03 L
Talendo® Extra	0.31 L		62 µL	4.14 µL	
Netzschwefel (Sulfur)	6 kg		1.2 g	80.16 mg	

Table S10: 4th Application – Day 31

Product	Helicopter Application per Hectare	Application in Spray Chamber (2 m^2)

	Amount of Product Used	Assumed Amount of Water Used	Amount of Product - Field Rate (100%)	Amount of Product - Drift Rate (6.68%)	Amount of Water
Luna® Experience	0.47 L	150 L	94 µL	6.28 µL	0.03 L
Flovine®	1.5 kg		0.3 g	20.04 mg	

Note that the fungicide *Zorvec Zelavin*® used in the 4th helicopter based application in 2023, is no longer manufactured and was therefore not included in the experiment.

Applications were performed in a spray chamber configured to simulate helicopter spraying in viticulture. The chamber was operated at a travel speed of 2.52 km h^{-1} and a spray pressure of 1.9 bar using an 8004E nozzle (Lechler GmbH, Germany). Separate spray solutions were prepared for control, drift, and field-rate treatments. Liquid formulations were pipetted directly into continuously stirred spray solutions, whereas solid formulations were weighed and added separately. Spray tanks were thoroughly rinsed between treatments to minimize cross-contamination.

Prior to each subsequent fungicide application, cups were transported to the laboratory and larvae were weighed. Dead individuals were removed, recorded, and stored at −20 °C. Larvae and host plants were then treated in the spray chamber following the procedure described above, using fresh plastic trays and newly collected *S. album* branches from the respective treatment. Pupae present at the time of application were also exposed. These were transferred together with the surrounding plant material and returned to their cups without additional food supply. As pupae remained partially covered by *S. album* during spraying, exposure was likely reduced, reflecting field conditions. Some larvae were excluded due to mechanical damage or loss during handling. The experiment was terminated shortly after the fourth and final application because a short-term temperature spike in the greenhouse resulted in high pupal mortality across treatments.

Supplementary Methods S6 – Detailed Methodology of the Luna® Experience Toxicity Experiments

Application of Luna® Experience followed the general spray protocol described for the Helicopter Spray Experiment. The field application rate documented for the Moselle Valley (0.47 L ha^{-1} in 150 L ha^{-1} water) was proportionally scaled to the spray chamber area as described in Supplementary Methods S5.

Individualised-Larvae Experiment

For the individualised-larvae experiment, 100 larvae were individually transferred to 35 mm plastic Petri dishes prior to treatment. Larvae were randomly assigned to the Luna® Experience or water-control treatment (n = 50 per treatment) and weighed before application using a precision scale. Larvae inside Petri dishes and corresponding trays containing rooted *S. album* were treated simultaneously in the spray chamber, with control treatments applied first. After spraying, larvae were transferred individually into 1 L transparent plastic cups without lids. Small perforations in the side walls and bottom ensured ventilation. Branches of treated *S. album* were

collected after application, pooled within treatments, and distributed evenly among cups. Cups were covered with insect mesh secured using rubber bands and randomly distributed within a climate chamber.
Climate chambers initially followed a diurnal temperature regime of 15–25 °C, 60% relative humidity, and a 16:8 h light:dark cycle based on meteorological data from a vineyard-associated weather station in the Moselle Valley (50.10581° N, 7.13259° E; https://www.wetter.rlp.de/Agrarmeteorologie/Wetterdaten/Mosel-Saar). This represents a known habitat of *P. a. vinningensis* and reflects typical late May to early June conditions. This period coincides with larval presence prior to pupation and overlaps with the usual application window of Luna® Experience. Relative humidity was later increased to 70% to reduce host plant desiccation.
Following increased baseline mortality, environmental conditions were standardized to commonly used *P. apollo* rearing conditions consisting of constant 28 °C, 70% relative humidity, and a 16:8 h light:dark cycle. A detailed timeline of climate chamber adjustments is provided in the tables below.

Supplementary Tables S11 to S15 – Climate chamber settings throughout the Luna® Experience Toxicity Experiment for individualised larvae (Tab. S11–14), pupae and adults (Tab. S15).

Table S11: Larvae - T0 (days)

Light	Time	Temeprature [°C]	Humidity [% RH]
ON (16 hrs)	6 AM - 9 AM	15	60
	9 AM - 11 AM	20	
	11 AM - 7 PM	25	
	7 PM - 9 PM	20	
	9 PM - 10 PM	15	
OFF (8 hrs)	10 PM - 6 AM	15	

Table S12: Larvae - T5 (days)

Light	Time	Temeprature [°C]	Humidity [% RH]
ON (16 hrs)	6 AM - 9 AM	15	65
	9 AM - 7 PM	25	
	7 PM - 9 PM	20	
	9 PM - 10 PM	15	
OFF (8 hrs)	10 PM - 6 AM	15	

Table S13: Larvae - T6 (days)

Light	Time	Temeprature [°C]	Humidity [% RH]
ON (16 hrs)	5 AM - 6 AM	15	70
	6 AM - 10 AM	20	
	10 AM - 6 PM	25	
	6 PM - 9 PM	20	
OFF (8 hrs)	9 PM - 5 AM	15	

Table S14: Larvae - T10 (days)

Light	Time	Temeprature [°C]	Humidity [% RH]
ON (16 hrs)	5 AM - 9 PM	28	70
OFF (8 hrs)	9 PM - 5 AM	28	

Table S15: Pupae + Adult

Light	Time	Temeprature [°C]	Humidity [% RH]
ON (16 hrs)	5 AM - 9 PM	28	80
OFF (8 hrs)	9 PM - 5 AM	28	

S. album trays were maintained in a greenhouse at 20 °C under artificial long-day conditions and watered beginning 48 h after treatment application. Food was replenished when plant material was consumed or began to desiccate. To standardize disturbance, fresh host plant material was added simultaneously across all experimental units, with quantities adjusted according to larval development.

Larvae were weighed 7, 13, and 19 days after application using a precision scale. Following each weighing event, cups were randomly repositioned within the climate chamber. Dead larvae were removed and stored individually at −20 °C. Cups containing newly formed pupae were transferred to a second climate chamber with identical temperature and light settings but elevated relative humidity (80%). A diagonally positioned cardboard-wrapped wire was added to each cup to facilitate climbing and wing expansion after adult emergence. Cups were checked daily for eclosed adults. Adults were allowed to harden for 24 h before being frozen individually at −20 °C in parchment bags, after which they were weighed and sexed.

Grouped-Larvae Experiment

The grouped-larvae experiment was conducted following high baseline mortality in the individualised-larvae trial. To provide a more structurally heterogeneous environment

and reduce handling disturbance, larvae were housed in larger group containers that allowed free movement and behavioral choice on host plants.

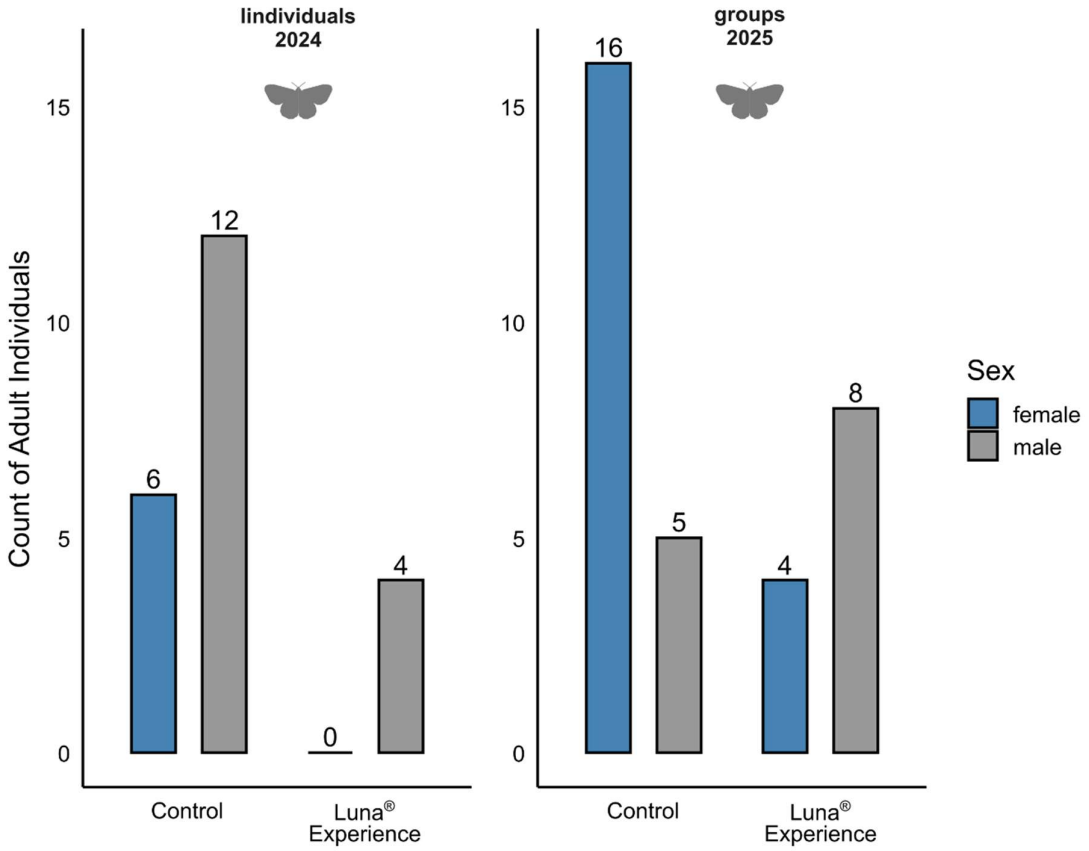
Larvae were assigned to larval stage classes based on head capsule morphology and randomly allocated to Luna® Experience or control treatments within each stage. Group sizes consisted of 10 L3 larvae, 50 L4 larvae, and 13 L5 larvae per treatment. To minimize handling stress, larval weight was not recorded during development. Larvae and corresponding *S. album* trays were treated simultaneously in the spray chamber following the same procedure described above. After application, larvae were transferred to 15 L plastic boxes (400 × 300 × 180 mm, L × W × H) without lids and lined with two layers of paper towels. Treated *S. album* branches were pooled within treatments and distributed evenly among boxes. Three sterilized slate stones (~50 × 20 mm) were placed centrally within each box to provide basking sites.

L3 and L5 larvae were maintained in one box per treatment, whereas L4 larvae were distributed across four boxes per treatment (two boxes containing 13 larvae and two containing 12 larvae). Boxes were covered with insect mesh and randomized across two identical climate chambers maintained at constant 28 °C, 40% relative humidity, and a 16:8 h light:dark cycle.

Freshly treated *S. album* trays were maintained in a greenhouse at approximately 20 °C under artificial long-day conditions. Because plants were supplied without roots, trays were covered with perforated transparent foil to maintain humidity while allowing air exchange. Food inside housing boxes was replenished every second day. Boxes were cleaned weekly unless larvae were undergoing pupation, during which paper towels and food substrate were replaced completely. Separate tools were used for each treatment and disinfected with ethanol between boxes to minimize cross-contamination. Dead larvae were removed and stored individually at −20 °C.

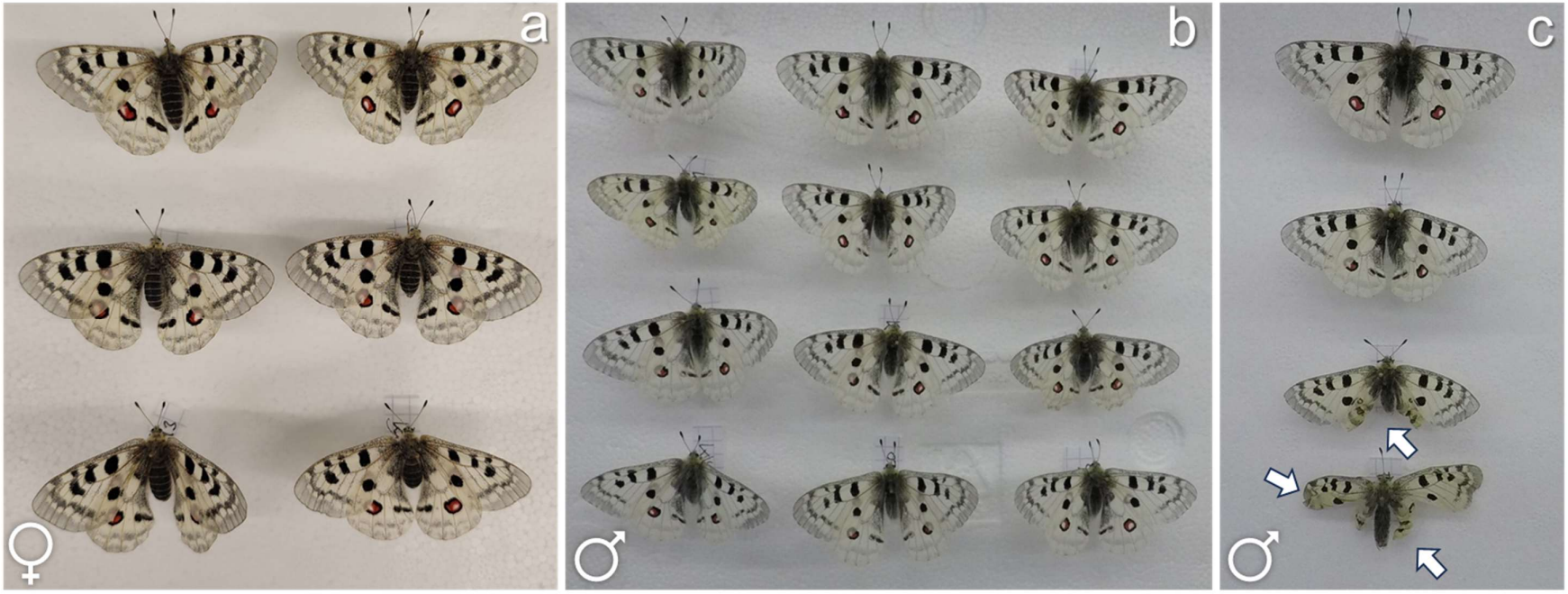
Newly formed pupae remained inside housing boxes until external wax structures had developed (1–2 days). Pupae were then transferred to a separate climate chamber (28 °C, 80% relative humidity, 16:8 h light:dark cycle) and placed inside mesh cages (300 × 300 × 300 mm) lined with paper towels. Crumpled paper towels were added to facilitate climbing after adult emergence. Adults were counted, allowed to harden for 24 h, frozen individually at −20 °C, weighed, and sexed.

Supplementary Figure S1 – Adult Sex Ratios in the Luna® Experience Toxicity Experiments



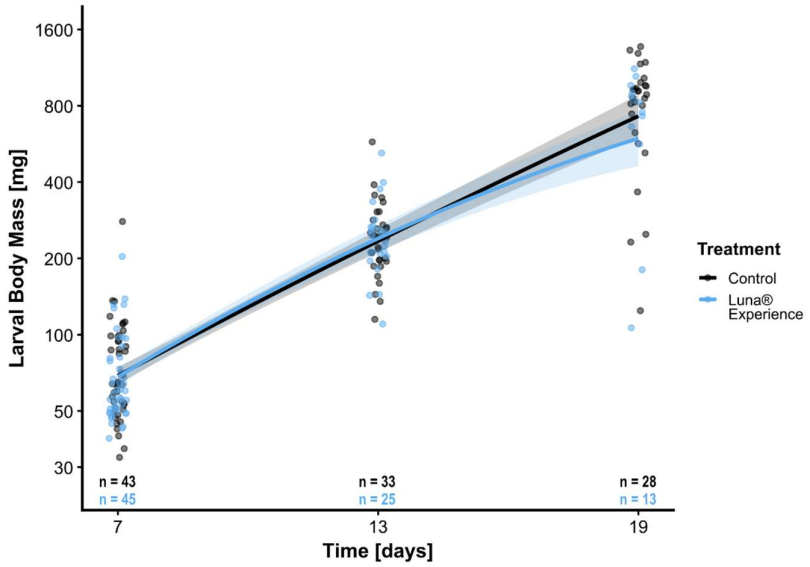
Supplementary Figure S1: Adult Sex Ratios in the Luna® Experience Toxicity Experiments for individually housed larvae (left) and larvae housed in groups (right).

Supplementary Figure S2 – Wing deformation in the Luna® Experience Toxicity Experiment



Supplementary Figure S2: Adults of *P. a. vinningensis* from the Luna® Experience Toxicity Experiment with individually housed larvae. Wings of two male specimen from the Luna® Experience treatment group showed strong deformation (c, white arrows), whereas wings of female and male specimen in the control group appeared normal (a & b).

Supplementary Figure S3 – Growth of *P. a. vinningensis* larvae in individual housing during the Luna® Experience toxicity experiment.



Suppl. Figure S2: Growth of *P. a. vinningensis* larvae in individual housing during the Luna® Experience toxicity experiment. Larvae and host plants were sprayed with fungicide (or water as a control) and allowed to feed on treated host plants in isolation while larval body mass was measured repeatedly. Growth trajectories depict model-predicted larval body mass over time (solid lines) based on a linear mixed-effects model. Shaded bands indicate 95% confidence intervals around the predicted means. Points represent individual observations from repeated measurements. n = number of biological replicates (i.e. number of larvae). The number of replicates decreased over time due to larval mortality.

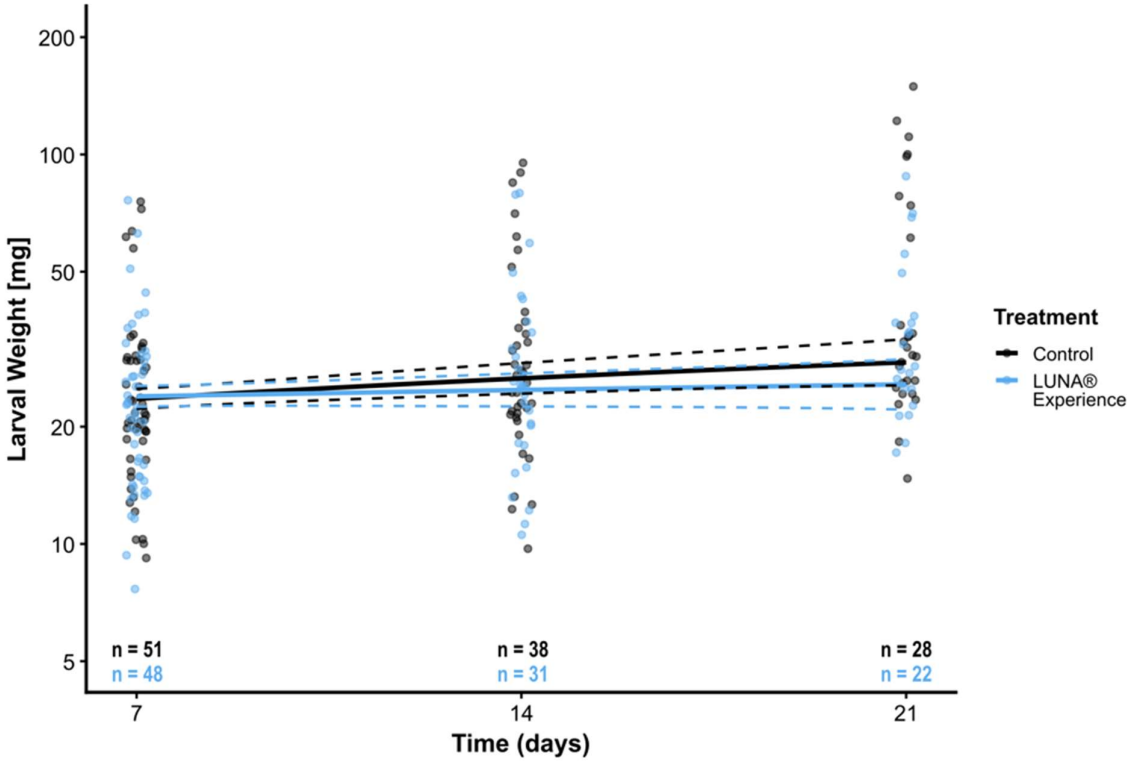
Supplementary Methods S7 – Terminated Luna® Experience Toxicity Experiment

A third Luna® Experience experiment using individually housed larvae was initiated following the same general protocol as the individualised-larvae experiment described in the main manuscript. However, the experiment was terminated prior to pupation because exceptionally high mortality occurred across both control and treatment groups.

Inspection of larval growth trajectories indicated unusually low body mass and delayed development from the onset of the experiment (Suppl. Fig. S3 below), suggesting that the cohort did not represent normal physiological conditions. Because this affected all treatments similarly, the experiment was not included in the main analyses. In the following, the experimental procedures and weight data collected prior to termination are provided (Suppl. Tab. S16).

Sedum album plants were collected from a remote, non-viticultural site in the Moselle valley in Dezember (50°10′27″ N, 7°11′51″ E). For propagation during the winter, plants

were grown in plastic trays containing a 1:1 (v/v) mixture of steamed leaf compost and washed sand in a greenhouse at 20 °C. Additional *Sedum* was collected on the institute's grounds (48°42′44″ N, 9°12′22″ E) and planted as well.
Approximately 200 *P. a. vinningensis* eggs were provided by the "Projekt Mosel-Apollo" (Stiftung für Natur und Umwelt im Landkreis Mayen-Koblenz). To initiate hatching, the eggs were placed inside plastic trays lined with kitchen paper and kept at 20 °C under a 16:8 light:dark cycle. Hatched larvae were transferred to several 5 L rectangular, transparent plastic boxes lined with kitchen paper. Freshly harvested *Sedum* branches were provided and the boxes were covered with insect mesh. Larval boxes and *Sedum* trays were transferred to rail-mounted platform trolleys in a greenhouse with winter-protection ventilation preventing temperatures from dropping below 10 °C. The greenhouse can be opened to move the trolleys outdoors. When outdoor temperatures exceeded 10 °C and conditions were favorable (i.e. no precipitation and wind), all trolleys were placed outdoors. In the evening, the trolleys were returned indoors before temperatures dropped to ≤ 10 °C. The larvae were provided with fresh *Sedum* plants ad libitum and were regularly transferred to clean boxes.



Supplementary Figure S3: Growth of *Parnassius apollo vinningensis* larvae in individual housing during the excluded Luna® Experience toxicity experiment. Larvae and host plants were sprayed with fungicide (or control) and allowed to feed on the respective host plants in isolation while larval weight was measured repeatedly. Growth trajectories depict model-predicted larval weight over time prior to experiment termination based on a linear mixed-effects model, with dashed lines indicating 95% CIs. Points represent individual observations from repeated measurements.

"n"=number of biological replicates (i.e. number of larvae). The number of replicates decreased over time due to larval mortality.

Supplementary Table S16 – Data collected in the terminated Luna® Experience trial

ID	treatment	sampling	days	time	start_weight	weight
1	Control	1	7	7	54,07	75,65
2	Control	1	7	7	7,98	20,6
3	Control	1	7	7	47,09	61,43
4	Control	1	7	7	45,5	63,53
6	Control	1	7	7	28,43	34,03
7	Control	1	7	7	26,61	31,69
8	Control	1	7	7	26,73	30,44
9	Control	1	7	7	23,94	29,72
10	Control	1	7	7	7,38	9,2
11	Control	1	7	7	22,05	25,67
12	Control	1	7	7	18,78	25,32
13	Control	1	7	7	18,65	19,94
14	Control	1	7	7	26,26	32,79
15	Control	1	7	7	7,53	10,01
17	Control	1	7	7	15,79	14,85
18	Control	1	7	7	9,18	12,76
19	Control	1	7	7	17,41	23,74
20	Control	1	7	7	20,71	29,53
21	Control	1	7	7	11,43	13,18
22	Control	1	7	7	16,98	22,89
23	Control	1	7	7	20,82	19,69
24	Control	1	7	7	22,16	34,5
25	Control	1	7	7	15,15	19,81
26	Control	1	7	7	36,08	57,41
27	Control	1	7	7	14,24	18,57
28	Control	1	7	7	11,29	16,53
29	Control	1	7	7	13,89	15,34
30	Control	1	7	7	16,76	32,14
32	Control	1	7	7	13,61	18,38
33	Control	1	7	7	13,99	19,48
34	Control	1	7	7	17,05	21,4
35	Control	1	7	7	14,05	20,84
36	Control	1	7	7	7,95	10,25
37	Control	1	7	7	7,2	10,28
39	Control	1	7	7	12,36	29,53
40	Control	1	7	7	13,41	16,43
41	Control	1	7	7	11,28	20,76
43	Control	1	7	7	8,78	21,67
44	Control	1	7	7	15,85	23,47

45	Control	1	7	7	13,98	19,69
46	Control	1	7	7	13,74	28,39
48	Control	1	7	7	12,82	20,44
49	Control	1	7	7	18,13	21,71
50	Control	1	7	7	13,95	29,62
51	Control	1	7	7	23,22	30,14
52	Control	1	7	7	15,63	23,26
53	Control	1	7	7	46,1	72,43
54	Control	1	7	7	12,29	19,42
55	Control	1	7	7	20,81	27,79
56	Control	1	7	7	7,83	12,1
57	Control	1	7	7	7,78	13,83
59	LUNA	1	7	7	16,44	25,36
60	LUNA	1	7	7	40	76,27
61	LUNA	1	7	7	7,44	7,66
63	LUNA	1	7	7	35,52	62,7
64	LUNA	1	7	7	40,17	50,88
65	LUNA	1	7	7	28	35,84
66	LUNA	1	7	7	12,35	16,61
67	LUNA	1	7	7	21,65	30,28
70	LUNA	1	7	7	27,08	36,74
71	LUNA	1	7	7	20,37	29,38
72	LUNA	1	7	7	19,5	27,22
73	LUNA	1	7	7	21,12	22,93
74	LUNA	1	7	7	16,04	22,41
75	LUNA	1	7	7	7,51	18,03
76	LUNA	1	7	7	17,64	26,07
77	LUNA	1	7	7	15,84	21,08
78	LUNA	1	7	7	20,56	26,86
79	LUNA	1	7	7	21,94	31,31
82	LUNA	1	7	7	16,68	28,11
83	LUNA	1	7	7	18,01	21,83
84	LUNA	1	7	7	9,13	14,51
85	LUNA	1	7	7	9,31	14,21
86	LUNA	1	7	7	10,71	16,27
87	LUNA	1	7	7	14,2	19,83
88	LUNA	1	7	7	9,06	13,37
89	LUNA	1	7	7	10,15	15,96
90	LUNA	1	7	7	9,24	13,71
91	LUNA	1	7	7	14,93	20,94
92	LUNA	1	7	7	20,68	38,74
93	LUNA	1	7	7	15,09	22,77
95	LUNA	1	7	7	13,68	19,38
96	LUNA	1	7	7	10,07	13,31
97	LUNA	1	7	7	15,43	20,56
98	LUNA	1	7	7	11,59	13,51
99	LUNA	1	7	7	11,44	14,92

100	LUNA	1	7	7	9,8	11,79
103	LUNA	1	7	7	8,19	9,35
104	LUNA	1	7	7	9,5	11,61
105	LUNA	1	7	7	16,5	26,93
106	LUNA	1	7	7	11,32	14
107	LUNA	1	7	7	11,85	14,85
108	LUNA	1	7	7	24,17	30,57
109	LUNA	1	7	7	17,12	24,04
110	LUNA	1	7	7	14,77	23,13
111	LUNA	1	7	7	22,31	32,77
112	LUNA	1	7	7	20,78	39,23
113	LUNA	1	7	7	7,96	13,04
114	LUNA	1	7	7	37,49	44,19
1	Control	2	14	14	54,07	95,13
2	Control	2	14	14	7,98	24,46
3	Control	2	14	14	47,09	70,47
6	Control	2	14	14	28,43	51,41
7	Control	2	14	14	26,61	31,36
8	Control	2	14	14	26,73	28,89
9	Control	2	14	14	23,94	27,86
11	Control	2	14	14	22,05	32,29
12	Control	2	14	14	18,78	25,94
14	Control	2	14	14	26,26	61,53
18	Control	2	14	14	9,18	12,28
19	Control	2	14	14	17,41	22,94
20	Control	2	14	14	20,71	27,09
22	Control	2	14	14	16,98	27,52
24	Control	2	14	14	22,16	37,21
26	Control	2	14	14	36,08	89,83
28	Control	2	14	14	11,29	16,57
30	Control	2	14	14	16,76	56,85
33	Control	2	14	14	13,99	22,5
34	Control	2	14	14	17,05	20,68
35	Control	2	14	14	14,05	21,65
36	Control	2	14	14	7,95	9,72
37	Control	2	14	14	7,2	13,22
39	Control	2	14	14	12,36	39,4
41	Control	2	14	14	11,28	21,09
43	Control	2	14	14	8,78	22,33
44	Control	2	14	14	15,85	17,03
45	Control	2	14	14	13,98	21,49
46	Control	2	14	14	13,74	35,83
48	Control	2	14	14	12,82	24,46
49	Control	2	14	14	18,13	21,91
50	Control	2	14	14	13,95	28,77
51	Control	2	14	14	23,22	34,59
53	Control	2	14	14	46,1	84,66

54	Control	2	14	14	12,29	23
55	Control	2	14	14	20,81	32,95
56	Control	2	14	14	7,83	12,62
57	Control	2	14	14	7,78	19,05
59	LUNA	2	14	14	16,44	23,03
60	LUNA	2	14	14	40	79,59
63	LUNA	2	14	14	35,52	78,87
64	LUNA	2	14	14	40,17	49,74
65	LUNA	2	14	14	28	37,25
66	LUNA	2	14	14	12,35	13,17
67	LUNA	2	14	14	21,65	28,25
70	LUNA	2	14	14	27,08	59,21
71	LUNA	2	14	14	20,37	30,22
72	LUNA	2	14	14	19,5	25,44
73	LUNA	2	14	14	21,12	22,02
74	LUNA	2	14	14	16,04	25,15
76	LUNA	2	14	14	17,64	25,53
78	LUNA	2	14	14	20,56	26,63
82	LUNA	2	14	14	16,68	31,76
83	LUNA	2	14	14	18,01	27,21
84	LUNA	2	14	14	9,13	20,14
86	LUNA	2	14	14	10,71	15,2
87	LUNA	2	14	14	14,2	17,91
88	LUNA	2	14	14	9,06	15,72
89	LUNA	2	14	14	10,15	18,15
92	LUNA	2	14	14	20,68	42,55
97	LUNA	2	14	14	15,43	20,37
100	LUNA	2	14	14	9,8	12,2
104	LUNA	2	14	14	9,5	11,25
105	LUNA	2	14	14	16,5	24,16
108	LUNA	2	14	14	24,17	34,89
109	LUNA	2	14	14	17,12	25,2
110	LUNA	2	14	14	14,77	26,13
113	LUNA	2	14	14	7,96	10,55
114	LUNA	2	14	14	37,49	43,32
1	Control	3	21	21	54,07	100,04
2	Control	3	21	21	7,98	22,84
3	Control	3	21	21	47,09	78,17
6	Control	3	21	21	28,43	61,11
7	Control	3	21	21	26,61	34,26
9	Control	3	21	21	23,94	29,28
11	Control	3	21	21	22,05	33,98
14	Control	3	21	21	26,26	73,95
20	Control	3	21	21	20,71	24,28
22	Control	3	21	21	16,98	28,59
24	Control	3	21	21	22,16	33,07
26	Control	3	21	21	36,08	98,81

28	Control	3	21	21	11,29	14,72
30	Control	3	21	21	16,76	121,97
33	Control	3	21	21	13,99	18,3
35	Control	3	21	21	14,05	26,26
39	Control	3	21	21	12,36	149,41
41	Control	3	21	21	11,28	25,9
43	Control	3	21	21	8,78	25,18
45	Control	3	21	21	13,98	30,52
46	Control	3	21	21	13,74	36,43
48	Control	3	21	21	12,82	31,95
50	Control	3	21	21	13,95	26,25
51	Control	3	21	21	23,22	34,7
53	Control	3	21	21	46,1	110,98
54	Control	3	21	21	12,29	23,45
55	Control	3	21	21	20,81	30,28
57	Control	3	21	21	7,78	24,26
59	LUNA	3	21	21	16,44	21,33
60	LUNA	3	21	21	40	55,54
63	LUNA	3	21	21	35,52	87,92
64	LUNA	3	21	21	40,17	70,54
65	LUNA	3	21	21	28	38,47
67	LUNA	3	21	21	21,65	27,47
70	LUNA	3	21	21	27,08	68,93
71	LUNA	3	21	21	20,37	35,07
74	LUNA	3	21	21	16,04	22,66
76	LUNA	3	21	21	17,64	27,99
82	LUNA	3	21	21	16,68	36,95
83	LUNA	3	21	21	18,01	25,1
84	LUNA	3	21	21	9,13	24,99
86	LUNA	3	21	21	10,71	17,16
88	LUNA	3	21	21	9,06	21,4
89	LUNA	3	21	21	10,15	18,16
92	LUNA	3	21	21	20,68	49,6
105	LUNA	3	21	21	16,5	26,52
108	LUNA	3	21	21	24,17	32,52
109	LUNA	3	21	21	17,12	28,58
110	LUNA	3	21	21	14,77	34,61
114	LUNA	3	21	21	37,49	36,9
1	Control	4	28	28	54,07	80,99
6	Control	4	28	28	28,43	63,81
11	Control	4	28	28	22,05	25,51
14	Control	4	28	28	26,26	61,33
26	Control	4	28	28	36,08	95
30	Control	4	28	28	16,76	280,64
39	Control	4	28	28	12,36	276,54
41	Control	4	28	28	11,28	22,26
43	Control	4	28	28	8,78	24,99

45	Control	4	28	28	13,98	27,22
46	Control	4	28	28	13,74	31,88
48	Control	4	28	28	12,82	31,25
53	Control	4	28	28	46,1	92,18
57	Control	4	28	28	7,78	22,57
64	LUNA	4	28	28	40,17	70,98
70	LUNA	4	28	28	27,08	76,09
82	LUNA	4	28	28	16,68	31,69
83	LUNA	4	28	28	18,01	22,6
84	LUNA	4	28	28	9,13	24,06
110	LUNA	4	28	28	14,77	29,07

Supplementary Analysis S1 – Determination of *Sedum album* Plant Nutrient Contents from the Field-Collected Host Plant Experiment and Larval Body Mass / Mode-of-Action Diversity Correlation Analysis

Sedum album plants collected at four locations in the Moselle valley and used in the Field-Collected Host Plant Experiment were analysed for plant nutrient contents and then correlated with final larval weight and the pesticide predictor mode of action (MoA) diversity.

Determination of Carbon and Nitrogen (CN Analysis)

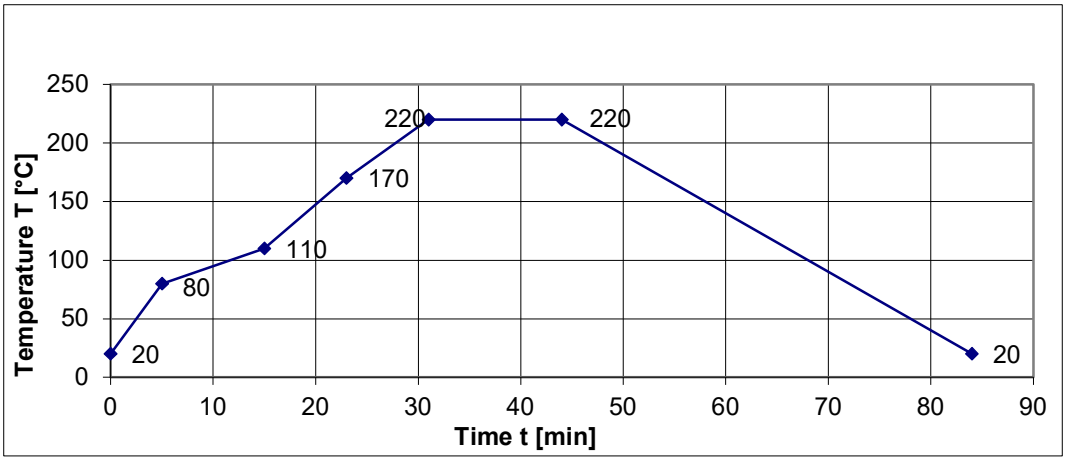
Carbon (C) and nitrogen (N) concentrations were determined using an elemental analyser (vario EL cube, Elementar Analysensysteme GmbH, Germany). Samples were combusted at high temperature (950 °C) in a stream of pure oxygen. During oxidative combustion, the elements C, H, and N were converted into CO_2, H_2O, NO, and NO_2, while molecular nitrogen (N_2) was also formed. If present, volatile halogen compounds were generated. Helium was used as carrier gas. Nitrogen oxides were quantitatively reduced to N_2 in a downstream copper reduction tube at 600 °C, where excess oxygen was simultaneously removed. Volatile halogen compounds were retained on silver wool and thus removed from the gas stream. Water vapor was removed using drying tubes filled with Sicapent. The remaining gases (CO_2 and N_2) were separated by gas chromatography using a CO_2 adsorption column and detected by a thermal conductivity detector (TCD). Quantification was based on peak integration, allowing simultaneous qualitative and quantitative determination of C and N. Element concentrations were calculated from detector response and sample mass. The method follows established protocols (e.g., VDLUFA Method Book, Vol. II.1, Method 3.5.2.7).

Microwave-Assisted Acid Digestion

Microwave-assisted acid digestion was used to decompose the sample matrix and bring elements into solution. Finely ground and homogenized plant material (~0.2 g) was placed in quartz vessels and mixed with 1 mL ultrapure water and 2.5 mL concentrated nitric acid (65%, Suprapur grade, Carl Roth, Germany). Digestion was

carried out in a closed microwave system under elevated pressure using a defined temperature program (Suppl. Fig. S4). During digestion, the sample matrix was decomposed and elements were released into solution.

The method follows standard procedures (e.g., VDLUFA Method Book, Vol. VII, Method 2.1.3).



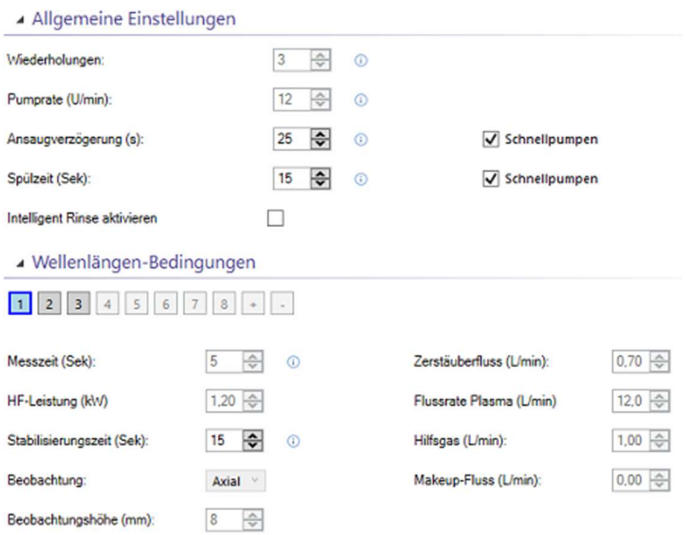
Supplementary Figure S4: Temperature program of the microwave digestion in the plant nutrient analysis.

Element Determination by ICP-OES

The concentrations of the elements Al, B, Ba, Ca, Cr, Cu, Fe, K, Mg, Mn, Na, P, S, and Zn were determined by inductively coupled plasma optical emission spectrometry (ICP-OES) using an Agilent 5110 (Agilent Technologies, Santa Clara, CA, USA). Sample solutions were nebulized to form an aerosol and introduced into an inductively coupled plasma, where the elements were atomized and excited. Emission lines were separated using a polychromator and detected at element-specific wavelengths. Wavelength selection was optimized with respect to matrix composition and potential spectral interferences. Measurements were performed in axial or radial view mode depending on sensitivity requirements. Quantification was carried out using external calibration with multi-element standard solutions (Inorganic Ventures, Christiansburg, VA, USA), typically applying linear calibration functions. Measurement time was approximately 3 min per sample, with a solution consumption of ~3 mL.

The method is based on established procedures (e.g., VDLUFA Method Book, Vol. VII, Method 2.2.2.6).

Device settings ICP-OES:



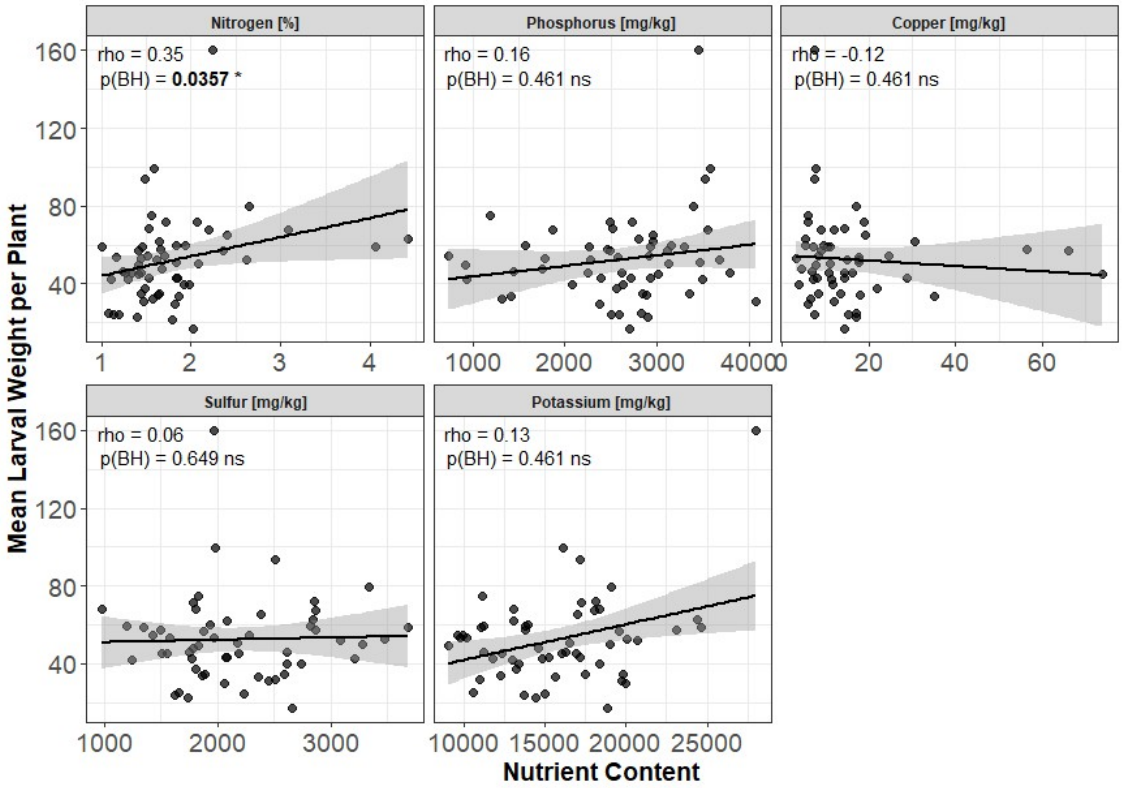
Correlation Analysis

To investigate if plant nutrients influenced larval weight after 31 days of exposure to host plants, a correlation analysis on the individual plant level was conducted for the following nutrients: nitrogen, phosphorus, copper, sulfur and potassium. Nitrogen was on the percentage scale, the other nutrients were given in mg/kg plant tissue. Larval weight was averaged per host plant. Relationships between nutrient content and mean larval weight were assessed using Spearman's rank correlation. To account for multiple testing across nutrients, p-values were adjusted using the Benjamini–Hochberg procedure. A positive significant correlation with larval weight was only detected for nitrogen (rho = 0.35, p = 0.036; Suppl. Tab. S17, Suppl. Fig. S5).

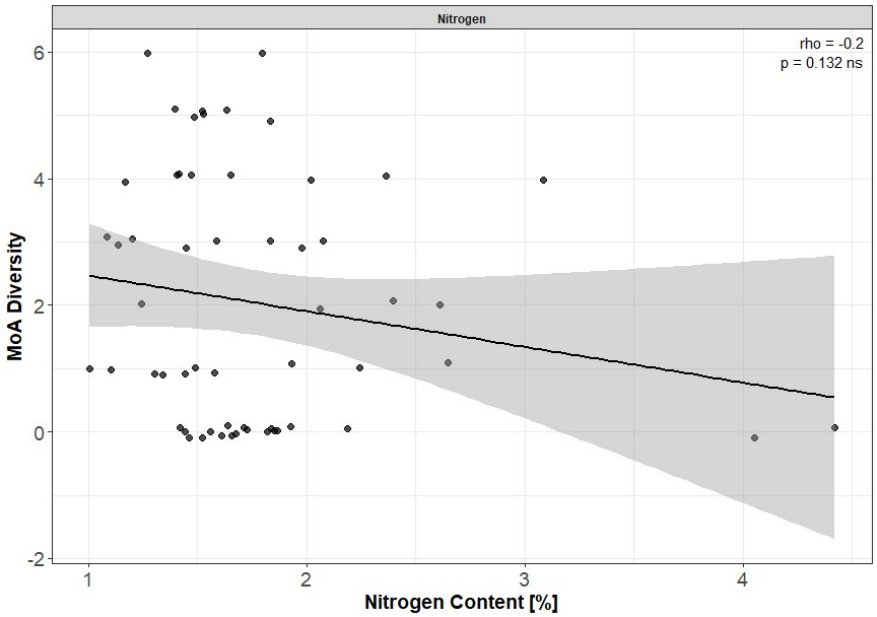
Supplementary Table S17: Results for the Spearman's rank correlation analysis between mean larval weight per host plant ID after 31 days of exposure and plant nutrient content.

nutrient	n	rho-value	p-value	p-value BH-adjusted
nitrogen	57	0.353	0.0071	0.0357
phosphorus	56	0.157	0.249	0.461
copper	56	-0.122	0.369	0.461
sulfur	56	0.062	0.649	0.649
potassium	56	0.128	0.346	0.461

In a separate analysis, the relationship between plant nitrogen content [%] and pesticide mode-of-action (MoA) diversity in host plants was examined using Spearman's correlation. This analysis revealed no evidence for an association between nitrogen and MoA diversity (rho = -0.2, p = 0.132; Suppl. Fig. S6).



Supplementary Figure S5: Relationship between plant nutrient content (nitrogen, phosphorus, copper, sulfur, potassium) and mean larval weight per plant after 31 days of exposure. Points represent individual plants, solid black lines show linear regressions with 95% confidence intervals. Spearman's rank correlation coefficients (rho) and Benjamini–Hochberg adjusted p-values are displayed in each panel with significant p-values in bold.



Supplementary Figure S6: Relationship between plant nitrogen content and mode-of-action diversity. Points represent individual plants, with linear regression

(solid black line) and 95% confidence interval. Spearman’s rank correlation coefficient (rho) and p-value are shown.

Statistical Report and Creation of Graphs

```
#########################################################################
##        Contaminated Host Plants – visualization of pesticide contents  -> Fig. 2            ##
#########################################################################
## Fig.2 panel a: Pesticide distribution and concentration per location
library(readxl)
library(dplyr)
library(ggplot2)
windowsFonts(Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial"))
df <- read_excel(".../Results_Sedum_final.xlsx",
                 sheet = "graph_pesticides per location")
str(df)
#########################################################################
library(ggpattern)
library(dplyr)
library(ggplot2)
library(scales)
library(ggnewscale)
library(colorspace)
# ----------------------------------
# 1) define herbicdes
# ----------------------------------
herbicides <- c(
  "Diflufenican",
  "Metamitron",
  "Metobromuron",
  "Propyzamide")
# ----------------------------------
# 2) Prep data
# ----------------------------------
```

```
df_plot <- df %>%
  rename( sum_conc = `sum concentration  [ng/g]`, moa_code = `MoA code`
  ) %>%
  mutate( moa_code = trimws(as.character(moa_code)),
    pesticide = as.character(pesticide)
  ) %>%
  filter(sum_conc > 0) %>%
  mutate(location = factor(
      location, levels = c("Dortebach", "Calmont", "Brauselay", "Rosenberg") ),
    pesticide_type = if_else(pesticide %in% herbicides, "Herbizid", "Fungizid")
  ) %>%
  group_by(location) %>%
  mutate(prop = sum_conc / sum(sum_conc)) %>%
  ungroup()
# ----------------------------------
# 3) Pestizid-summary
# ----------------------------------
pest_summary <- df_plot %>%
  group_by(pesticide, moa_code, pesticide_type) %>%
  summarise(total_conc = sum(sum_conc), .groups = "drop")
# ----------------------------------
# 4) MoA-order for fungicdes
#    per summed totalconc.
# ----------------------------------
fung_moa_order <- pest_summary %>%
  filter(pesticide_type == "Fungizid") %>%
  group_by(moa_code) %>%
  summarise(moa_total_conc = sum(total_conc), .groups = "drop") %>%
  arrange(desc(moa_total_conc)) %>%
  pull(moa_code)
# ----------------------------------
```

```
# 5) Pesticide Order
# ----------------------------------
# Fungizides:
# - per MoA-Group
# - MoA-Gruppen nach Gesamt-Konzentration
# - innerhalb MoA nach absteigender Gesamtkonzentration
fungicide_levels <- pest_summary %>%
  filter(pesticide_type == "Fungizid") %>%
  mutate(moa_code = factor(moa_code, levels = fung_moa_order)) %>%
  arrange(moa_code, desc(total_conc), pesticide) %>%
  pull(pesticide)

# Herbizides:
# - explicit order:
herbicide_priority <- c("Diflufenican", "Propyzamide", "Metamitron", "Metobromuron")
herbicide_levels <- herbicide_priority[herbicide_priority %in% pest_summary$pesticide]
other_herbicides <- pest_summary %>%
  filter(pesticide_type == "Herbizid", !pesticide %in% herbicide_levels) %>%
  arrange(desc(total_conc), pesticide) %>%
  pull(pesticide)

herbicide_levels <- c(herbicide_levels, other_herbicides)
# total order for plot
pesticide_levels <- c(fungicide_levels, herbicide_levels)
df_plot <- df_plot %>%
  mutate(    pesticide = factor(pesticide, levels = pesticide_levels)  )
# ----------------------------------
# 6) Colours for pesticides
# ----------------------------------
pesticide_colors <- c(
  "Fluopyram"      = "#015EB2",
```

```
  "Fluxapyroxad"   = "#2E9DFD",
  "Boscalid"       = "#85C2FD",
  "Cyazofamid"     = "#28CDCA",
  "Fluopicolide"   = "#FEDF32",
  "Metrafenone"    = "#AF6EB1",
  "Dimethomorph"   = "#632279",
  "Cyflufenamid"   = "#4E4E4D",
  "Difenoconazole" = "#048F3E",
  "Penconazole"    = "#3CC26A",
  "Myclobutanil"   = "#A2DFAA",
  "Spiroxamine"    = "#F06796",
  "Cyprodinil"     = "#EC3742",
  "Diflufenican"   = "#935C1E",
  "Propyzamide"    = "#E2BE90")
# ----------------------------------
# 8) Labels for legends
# ----------------------------------
moa_lookup <- pest_summary %>%
  mutate(label = paste0(pesticide, " (", moa_code, ")")  )
label_vec <- moa_lookup$label
names(label_vec) <- moa_lookup$pesticide
# ----------------------------------
# 9) Order within legends
# ----------------------------------
legend_order_fung <- fungicide_levels
legend_order_herb <- herbicide_levels
fungicide_cols <- pesticide_colors[legend_order_fung]
herbicide_cols <- pesticide_colors[legend_order_herb]
# ----------------------------------
# 10) Coordiunates for hovering bars
# ----------------------------------
```

```
offset <- 0.02

df_rect <- df_plot %>%

  arrange(location, pesticide) %>%

  group_by(location) %>%

  mutate(  ymin = offset + lag(cumsum(prop), default = 0),

    ymax = offset + cumsum(prop),  ymid = (ymin + ymax) / 2,  x = as.numeric(location),  xmin = x - 0.38,  xmax = x + 0.38,  label = if_else(prop >= 0.05, percent(prop, accuracy = 1), "")

  ) %>%

  ungroup()

df_rect_fung <- df_rect %>%

  filter(pesticide_type == "Fungizid")

df_rect_herb <- df_rect %>%

  filter(pesticide_type == "Herbizid")

# ----------------------------------

# 10b) Total Conc. per location  for side-bars

# ----------------------------------

location_totals <- df_plot %>% group_by(location) %>% summarise(total_conc = sum(sum_conc), .groups = "drop") %>%

  mutate( x = c(1, 2, 3, 4),  xmin = x + 0.42,  xmax = x + 0.54 )

max_total_conc <- max(location_totals$total_conc)

location_totals <- location_totals %>%

  mutate( ymin = offset, ymax = offset + total_conc / max_total_conc )

# ----------------------------------

# 11) Plot

# ----------------------------------

p_current <- ggplot() +

  geom_rect( data = df_rect_fung, aes( xmin = xmin, xmax = xmax,  ymin = ymin, ymax = ymax,

     fill = pesticide ), color = "white", linewidth = 0.35

  ) +

  scale_fill_manual(values = fungicide_cols, breaks = legend_order_fung,

   labels = unname(label_vec[legend_order_fung]), name = "Fungicides", guide = guide_legend(order = 1)
```

```
  ) +
  ggnewscale::new_scale_fill() +
  geom_rect( data = df_rect_herb, aes( xmin = xmin, xmax = xmax,  ymin = ymin, ymax = ymax,
fill = pesticide ), color = "white",
    linewidth = 0.35 ) +
  scale_fill_manual( values = herbicide_cols, breaks = legend_order_herb,
    labels = unname(label_vec[legend_order_herb]), name = "Herbicides",
    guide = guide_legend(order = 2) ) +
  # Nebenbalken für absolute Konzentration
  geom_rect_pattern( data = location_totals, aes( xmin = xmin, xmax = xmax,
      ymin = ymin, ymax = ymax ), pattern = "stripe",  pattern_angle = 45,  pattern_density = 0.5,
pattern_spacing = 0.01, alpha = 0.7,  color = "black",
    linewidth = 0.6) +
  geom_text( data = df_rect, aes( x = x, y = ymid,  label = label),
    size = 3.4, fontface = "bold") +
  scale_x_continuous(breaks = 1:4, labels = levels(df_plot$location),
    expand = expansion(mult = c(0.06, 0.06))
  )+
  scale_y_continuous( breaks = offset + c(0, 0.25, 0.50, 0.75, 1.00),
    labels = c("0", "25", "50", "75", "100"), limits = c(0, offset + 1.02),
    expand = c(0, 0), sec.axis = sec_axis( transform = ~ (. - offset) * max_total_conc,
      breaks = unique(c( pretty(c(0, max_total_conc), n = 4), max_total_conc )),
      labels = scales::label_number(accuracy = 1),
      name = "Summed Pesticide Concentration [µg/kg]") ) +
  labs( x = "Location", y = "Relative Contribution to Summed Pesticide Concentration [%]"
  ) +
  theme_classic(base_size = 13) +
  theme(text = element_text(family = "Arial"),
    axis.line = element_line(linewidth = 1.0, color = "black"),
    axis.ticks = element_line(linewidth = 1.0, color = "black"),
    theme( axis.ticks.length = unit(-2, "pt") ),
    axis.text.x = element_text(
```

```
    angle = 45, hjust = 1, face = "bold", size = 11),
  axis.text.y = element_text(
    face = "bold", size = 11),
  axis.text.y.right = element_text(
    face = "bold", size = 11),
  axis.title.x = element_text(
    face = "bold",  size = 13),
  axis.title.y = element_text(
    face = "bold",  size = 13),
  axis.title.y.right = element_text(
    face = "bold",  size = 13),
  legend.title = element_text(face = "bold", size = 11),
  legend.text = element_text(size = 10),
  panel.grid = element_blank())
p_current

## Fig.2 panel a: Pesticide distribution and concentration per location
df <- read_excel("....xlsx", sheet = "growth")
df$ID_running=as.factor(df$ID_running)
df$location=as.factor(df$location)
str(df)
####################
df2 <- read_excel("C....xlsx", sheet = "pest.parameters")

df2$location=as.factor(df2$location)
df2$ID_running=as.factor(df2$ID_running)
str(df2)
########################
# combined plot
ibrary(tidyverse)
library(ggplot2)
```

```
install.packages("ggh4x")
library(ggh4x)
# ------------------------------------------------------------
# BASELINE DATA: one row per ID (robust)
# If df already includes repeated measures, we take first record per ID.
# For df2 (58 IDs), it's already one row per ID.
# ------------------------------------------------------------
# A) Indices from df (presence/absence FRAC/HRAC columns)
df_idx_s1 <- df %>%
  arrange(ID_running, sampling) %>%
  group_by(ID_running) %>%
  slice(1) %>%
  ungroup() %>%
  mutate(
    ID_running = as.factor(ID_running),
    location = as.factor(location),
    fungal_load = FRAC_B + FRAC_C + FRAC_D + FRAC_G + FRAC_H + FRAC_U,
    herbicide_load = HRAC_PDS + HRAC_PS2 + HRAC_MT,
    stress_load = fungal_load + herbicide_load
  ) %>%
  select(ID_running, location, fungal_load, herbicide_load, stress_load)

# B) Concentration + npest from df2 (already one row per ID)
df_pest <- df2 %>%
  mutate(
    ID_running = as.factor(ID_running),
    location = as.factor(location),
    sumconpest_log1p = log1p(sumconpest)
  ) %>%
  select(ID_running, location, npest, sumconpest_log1p)
# ------------------------------------------------------------
```

```
# LONG FORMAT for facetted plotting
# ----------------------------------------------------------
plot_idx <- df_idx_s1 %>%
  pivot_longer( cols = c(fungal_load, herbicide_load, stress_load),
    names_to = "metric", values_to = "value"
  ) %>%
  mutate(metric = recode( metric,
    fungal_load = "Fungicide MoA Diversity",
    herbicide_load = "Herbicide MoA Diversity",
    stress_load = "Total MoA Diversity" ))

plot_pest <- df_pest %>%
  pivot_longer( cols = c(npest, sumconpest_log1p),
    names_to = "metric", values_to = "value"
  ) %>%
  mutate(metric = recode( metric, npest = "Number of Pesticides",
    sumconpest_log1p = "Total Pesticide Concentration (log (1+ ng g⁻¹))"
  ))

plot_all <- bind_rows(plot_idx, plot_pest)

# Reihenfolge der Panels
plot_all$metric <- factor(plot_all$metric,
  levels = c( "Total Pesticide Concentration (log (1+ ng g⁻¹))",
    "Number of Pesticides", "Herbicide MoA Diversity", "Fungicide MoA Diversity",
    "Total MoA Diversity"))
# ----------------------------------------------------------
# PLOT
# ----------------------------------------------------------
plot_all <- plot_all %>%
  mutate(location = factor( location,  levels = c("Dortebach", "Calmont", "Brauselay",
"Rosenberg")
```

```
  ))

p <- ggplot(plot_all, aes(x = location, y = value)) +

  # Boxplots
  geom_boxplot(  width = 0.55,  outlier.shape = NA, linewidth = 0.6
  ) +
    # Einzelwerte
  ggbeeswarm::geom_quasirandom(
    width = 0.20,  alpha = 0.85,  size = 1.7, shape = 16
  ) +
    # Panels
  facet_wrap( ~ metric, scales = "free_y",  ncol = 1
  ) +
    # Spezifische y-Achsen
  ggh4x::facetted_pos_scales(
    y = list(  metric == "Number of Pesticides" ~
        scale_y_continuous( breaks = c(0, 3, 6, 9), limits = c(0, 9), expand = expansion(mult =
c(0, 0.05))
        ),
           metric == "Herbicide MoA Diversity" ~
        scale_y_continuous(   breaks = c(0, 1), limits = c(0, 1), expand = expansion(mult = c(0,
0.05))
 ) )
  ) +
    # Gemeinsames y-Label, kein x-Label
  labs( x = NULL,  y = "Value"
  ) +
    theme_classic(base_size = 11) +
    theme( # Schriftart
    text = element_text(family = "Arial"),
        # Panelüberschriften
```

```
    strip.background = element_blank(),
    strip.text = element_text( family = "Arial", face = "bold", color = "#3333CC", size = 10.5, margin = margin(t = 1, b = 2) ),
      # Standortnamen
    axis.text.x = element_text( family = "Arial",  angle = 25,  hjust = 1,  size = 10.5, face = "bold" ),
      # y-Achsenwerte
    axis.text.y = element_text( family = "Arial",  size = 10 ),
      # gemeinsame y-Achsenüberschrift
    axis.title.y = element_text( family = "Arial", size = 11, face = "bold",  margin = margin(r = 8)),

    # x-Achsentitel entfernen
    axis.title.x = element_blank(),
    # Achsen
    axis.line = element_line(
      linewidth = 0.75, color = "black"
    ),
      axis.ticks = element_line(
      linewidth = 0.6, color = "black"
    ),
      axis.ticks.length = unit(2.5, "pt"),
      # Abstand zwischen Panels
    panel.spacing.y = unit(0.20, "lines"),
      # Außenabstände
    plot.margin = margin( t = 5, r = 5, b = 5,  l = 5 ) )
p



 ##########################################################################
 ##   Contaminated Host Plants – larval weight -> TOTAL PESTICIDE CONCENTRATION  ##
 ##########################################################################

##  1. growth models / 2. Final weight models ##
```

```
# 1. growth models ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
library(readxl)
library(lme4)
library(dplyr)
windowsFonts(Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

df <- read_excel("data_weight_aufzuchtversuch.xlsx", sheet = "growth")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
df$sumconpest=as.numeric(df$sumconpest)
df$weight=as.numeric(df$weight)
str(df)
any(is.na(df))
colSums(is.na(df))

# data prep:
df <- df %>%
  filter(
    !is.na(weight),
    !is.na(sumconpest)   # sumconpest = NA for some observations -> exclude to allow same n
for null- & pesticide models
  ) %>%
  mutate(
    time_c = time - mean(time, na.rm = TRUE),     # time centered -> -7, 0, +7:
    sumconpest_z = as.numeric(scale(sumconpest))) # scale pesticide predictor (z-score)

colSums(is.na(df))

## systematic model comparison:
# NOTE: REML=FALSE because fixed effects differ between models -> ML instead
library(lme4)
library(lmerTest)
library(dplyr)

## models:
# NOTE: for AIC comparison: REML=FALSE because fixed effects differ between models ->
ML instead

# singular fit for all models including (1 + time_c | ID_running) structure
#   -> models can't estimate variance in growth slopes for individual cups

# Null model (location + time only):
mSC0_null <- lmer(log(weight) ~ location + time_c + I(time_c^2) +
    (1 | ID_running), data = df, REML = FALSE)
```

```
# pesticide models:
mSC1_linear <- lmer(log(weight) ~ location + sumconpest_z +
   time_c + I(time_c^2) + (1 | ID_running), data = df, REML = FALSE)

mSC2_quad <- lmer(log(weight) ~ location + sumconpest_z +cart+ I(sumconpest_z^2) +
   time_c + I(time_c^2) + (1 | ID_running),data = df, REML = FALSE)

mSC3_int_time <- lmer(log(weight) ~ location + sumconpest_z * (time_c + I(time_c^2)) +
   (1 | ID_running),data = df, REML = FALSE)

nobs_vec <- c(SC0_null = nobs(mSC0_null),  SC1_lin  = nobs(mSC1_linear),  SC2_quad =
nobs(mSC2_quad),  SC3_time = nobs(mSC3_int_time))
print(nobs_vec) # n = 421 for all models -> AIC comp. valid

# AIC comparison:
model_list_growth_sc <- list(
  SC0_null = mSC0_null,
  SC1_lin  = mSC1_linear,
  SC2_quad = mSC2_quad,
  SC3_time = mSC3_int_time)

aic_growth_sc <- sapply(model_list_growth_sc, AIC)

aic_growth_sc_df <- data.frame(
  Model = names(aic_growth_sc),
  AIC   = as.numeric(aic_growth_sc)
) %>%
  arrange(AIC) %>%
  mutate(dAIC = AIC - min(AIC))

print(aic_growth_sc_df)
library(performance)
library(see)
check_model(mSC0_null)

mSC1_linear <- update(mSC1_linear, REML=TRUE)
summary(mSC1_linear)
confint(mSC1_linear, parm = "sumconpest_z")

## 2. final weight models ++++++++++++++++++++++++++++++++++++++++++++++++++

df <- read_excel("data_weight_aufzuchtversuch.xlsx",
                 sheet = "final_weight")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
df$sumconpest=as.numeric(df$sumconpest)
```

```
str(df)
any(is.na(df))
colSums(is.na(df))

# data prep:
df <- df %>%
  filter(!is.na(sumconpest)  # sumconpest = NA for some observations -> exclude to allow same
n for null- & pesticide models)
colSums(is.na(df))

df <- df %>% mutate(sumconpest_z = as.numeric(scale(sumconpest)))

# Null model (location only)
mSCF0_null <- lmer(  weight ~ location + (1|ID_running),
  data = df, REML=FALSE)

# pesticide models:
mSCF1_linear <- lmer(  weight ~ location + sumconpest_z + (1|ID_running),
  data = df, REML=FALSE)
mSCF2_quad  <-  lmer(    weight  ~  location  +  sumconpest_z  +  I(sumconpest_z^2)  +
(1|ID_running),
  data = df, REML=FALSE)

# AIC comparison:
model_list_final_sc <- list(
  SCF0_null = mSCF0_null,
  SCF1_lin  = mSCF1_linear,
  SCF2_quad = mSCF2_quad)

aic_final_sc <- sapply(model_list_final_sc, AIC)

aic_final_sc_df <- data.frame(
  Model = names(aic_final_sc),
  AIC   = as.numeric(aic_final_sc)
) %>%
  arrange(AIC) %>%
  mutate(dAIC = AIC - min(AIC))
print(aic_final_sc_df) # null-model is best supported

mSCF1_linear <- update(mSCF1_linear, REML=TRUE)

summary(mSCF1_linear) # linear pest. model within delta-AIC <2, but sumconpest n.s.
library(performance)
library(see)
check_model(mSCF1_linear)
check_model(mSCF0_null)
range(df$sumconpest_z)
sd(df$sumconpest, na.rm = TRUE)
```

```
      ###################################################################
 ++++##   Contaminated Host Plants – larval weight -> NUMBER OF PESTICIDES   ##++++
      ###################################################################

# 1. npest as numeric
#     1.1 - growth (all samplings)
#     1.2 - final weight (31 days)

# 2. npest categorical
#     2.1 - growth (all samplings)
#     2.2 - final weight (31 days)


## 1.1 npest as numeric ->  growth (all samplings) ++++++++++++++++++++++++++++++++

library(readxl)
library(dplyr)
windowsFonts(  Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

df <- read_excel("data_weight_aufzuchtversuch.xlsx", sheet = "growth")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
df$npest=as.numeric(df$npest)
df$weight=as.numeric(df$weight)
str(df)
any(is.na(df))
colSums(is.na(df))

## systematic model comparison - global growth model
library(lme4)
library(lmerTest)
library(dplyr)

# time centered -> -7, 0, +7:
df <- df %>%
  mutate(time_c = time - mean(time, na.rm = TRUE))
str(df)

# Nullmodell (baseline)
m_np0_null <- lmer(log(weight) ~ location + time_c + I(time_c^2) + (1 | ID_running),
  data = df, REML=FALSE)

# npest as main effect only (no effect on growth rate)
```

```
m_np1_main <- lmer(log(weight) ~ location + npest + time_c + I(time_c^2) +
    (1 | ID_running), data = df, REML=FALSE)

# npest × time (linear growth only)
m_np2_time_linear <- lmer(log(weight) ~ location + npest * time_c + I(time_c^2) + (1 |
ID_running),
  data = df, REML=FALSE)

# npest × (time + time^2) - biologically plausible model
m_np3_full <- lmer(log(weight) ~ location + npest * (time_c + I(time_c^2)) +
    (1 | ID_running), data = df, REML=FALSE)

# npest × quadratic term only (effect on curvature only)
m_np4_curve_only <- lmer(log(weight) ~ location + time_c + I(time_c^2) +
    npest:I(time_c^2) + (1 | ID_running), data = df, REML=FALSE)

# 2. AIC comparison:
model_list_npest <- list(
  np0_null         = m_np0_null,
  np1_main         = m_np1_main,
  np2_time_linear  = m_np2_time_linear,
  np3_full         = m_np3_full,
  np4_curve_only   = m_np4_curve_only)

# extraction of AICs
aic_vec_npest <- sapply(model_list_npest, AIC)

aic_npest_df <- data.frame(
  Model = names(aic_vec_npest),
  AIC   = as.numeric(aic_vec_npest))

# sort & compute delta AIC
aic_npest_df <- aic_npest_df[order(aic_npest_df$AIC), ]
aic_npest_df$dAIC <- aic_npest_df$AIC - min(aic_npest_df$AIC)

print(aic_npest_df) # --> 0-model within delta AIC < 2

m_np4_curve_only <- update(m_np4_curve_only, REML=TRUE)
summary(m_np4_curve_only)
confint(m_np4_curve_only, parm = "I(time_c^2):npest")

m_np1_main <- update(m_np1_main, REML=TRUE)
summary(m_np1_main)
confint(m_np1_main, parm = "npest")

## 1.2 npest as numeric -> final weight (31 days) ++++++++++++++++++++++++++++++++

df <- read_excel("data_weight_aufzuchtversuch.xlsx",
```

```
                sheet = "final_weight")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
df$npest=as.numeric(df$npest)
any(is.na(df))
colSums(is.na(df))
str(df)

# null model (location only)
mFn0_null <- lmer(weight ~ location + (1|ID_running),
  data = df, REML = FALSE)

# pesticide models:
mFn1_npest_main <- lmer(weight ~ location + npest + (1|ID_running),
  data = df, REML = FALSE)

mFn2_npest_quad <- lmer(weight ~ location + npest + I(npest^2) + (1|ID_running),
  data = df, REML = FALSE)

# AIC comparison (Final Weight)
model_list_npest_last <- list(
  F0_null       = mFn0_null,
  F1_npest_main  = mFn1_npest_main,
  F2_npest_quad  = mFn2_npest_quad)

# extract AIC
aic_vec_last <- sapply(model_list_npest_last, AIC)

aic_last_df <- data.frame(
  Model = names(aic_vec_last),
  AIC   = as.numeric(aic_vec_last))

# sort & calculate delta AIC
aic_last_df <- aic_last_df[order(aic_last_df$AIC), ]
aic_last_df$dAIC <- aic_last_df$AIC - min(aic_last_df$AIC)
print(aic_last_df) # all within delta AIC < 2

mFn1_npest_main <- update(mFn1_npest_main, REML=TRUE)
summary(mFn1_npest_main)
confint(mFn1_npest_main, parm = "npest")

## 2.1 npest categorical -> growth (all samplings) ++++++++++++++++++++++++++++++++

library(readxl)
library(lme4)
library(dplyr)
```

```
windowsFonts(  Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

df <- read_excel("data_weight_aufzuchtversuch.xlsx", sheet = "growth")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
df$npest=as.numeric(df$npest)
df$weight=as.numeric(df$weight)
str(df)
any(is.na(df))
colSums(is.na(df))

## Categorization:
table(df$npest) # n for npest (all samplings!)

df_counts <- df %>%
  group_by(ID_running) %>%
  summarise(
    npest = unique(npest),
    n_measurements = n())
table(df_counts$npest) # n for npest (without repeated measures)

## categories for growth model:
df$npest_cat <- cut(
  df$npest,
  breaks = c(-Inf, 0, 1, 4, 7, Inf), # define borders of categories
  labels = c("0", "1", "2-4", "5-7", "8-9"))

## systematic model comparison - global growth model – categorical:

# time centered -> -7, 0, +7:
df <- df %>%
  mutate(time_c = time - mean(time, na.rm = TRUE))
str(df)

# Null model (location + time only)
mGc0_null <- lmer(log(weight) ~ location + time_c + I(time_c^2) +
    (1 | ID_running),data = df, REML = FALSE)

# npest_cat as additive effect
mGc1_additive <- lmer(log(weight) ~ location + npest_cat +
    time_c + I(time_c^2) + (1 | ID_running),  data = df, REML = FALSE)

# Interaction with linear time
mGc2_int_time <- lmer(log(weight) ~ location + npest_cat * time_c +
    I(time_c^2) + (1 | ID_running),  data = df, REML = FALSE)
```

```
# Interaction with linear + quadratic time
mGc3_int_time2 <- lmer(log(weight) ~ location + npest_cat * (time_c + I(time_c^2)) +
    (1 | ID_running),  data = df, REML = FALSE)

# AIC comparison:
# model list
model_list_growth_cat <- list(
  Gc0_null       = mGc0_null,
  Gc1_additive   = mGc1_additive,
  Gc2_time       = mGc2_int_time,
  Gc3_time2      = mGc3_int_time2)

aic_growth_vec <- sapply(model_list_growth_cat, AIC)

aic_growth_df <- data.frame(
  Model = names(aic_growth_vec),
  AIC = as.numeric(aic_growth_vec))

aic_growth_df <- aic_growth_df[order(aic_growth_df$AIC), ]
aic_growth_df$dAIC <- aic_growth_df$AIC - min(aic_growth_df$AIC)
print(aic_growth_df) # null-model best supported

## 2.2 npest categorical -> final weight (31 days) ++++++++++++++++++++++++++++++++

library(readxl)
library(lme4)
library(ggplot2)
library(dplyr)
windowsFonts(Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

df <- read_excel("data_weight_aufzuchtversuch.xlsx",
                 sheet = "final_weight")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
df$npest=as.numeric(df$npest)
str(df)
any(is.na(df))
colSums(is.na(df))

# categorization:
df$npest_cat <- cut(
  df$npest,
  breaks = c(-Inf, 0, 1, 4, 7, Inf), # define borders of categories
  labels = c("0", "1", "2-4", "5-7", "8-9"))
```

```
# Null model:
mFc0_null <- lmer(weight ~ location + (1|ID_running),
  data = df, REML = FALSE)

# pesticide models:
mFc1_additive <- lmer(weight ~ location + npest_cat + (1|ID_running),
  data = df, REML = FALSE)

mFc2_ordered <- lmer(weight ~ location + as.numeric(npest_cat) + (1|ID_running),
  data = df, REML = FALSE)

# AIC comparison:
model_list_final_cat <- list(
  Fc0_null      = mFc0_null,
  Fc1_additive  = mFc1_additive,
  Fc2_ordered   = mFc2_ordered)

aic_final_vec <- sapply(model_list_final_cat, AIC)

aic_final_df <- data.frame(
  Model = names(aic_final_vec),
  AIC = as.numeric(aic_final_vec))

aic_final_df <- aic_final_df[order(aic_final_df$AIC), ]
aic_final_df$dAIC <- aic_final_df$AIC - min(aic_final_df$AIC)
print(aic_final_df) # all within delta AIC <2

mFc2_ordered <- update(mFc2_ordered, REML=TRUE)
summary(mFc2_ordered)
confint(mFc2_ordered, parm = "as.numeric(npest_cat)")

            ##########################################################
 +++++++++##   Contaminated Host Plants – larval weight -> MoA GROUPS   ##+++++++++
            ##########################################################

# 1. FRAC/HRAC individual (binary)
#     - growth (all samplings)
#     - final weight (31 days)

# 2. FRAC/HRAC indices (FRAC_load, HRAC_load, Combined_load)
#     - growth (all samplings)
#     - final weight (31 days)


#  data prep ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

library(readxl)
library(dplyr)
```

```
windowsFonts(Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial"))

df <- read_excel("data_weight_aufzuchtversuch.xlsx", sheet = "growth")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
str(df)
any(is.na(df))
colSums(is.na(df))

## 1.1 FRAC/HRAC growth model (all samplings) - global (across locations) ++++++++++

library(tidyverse)
library(lme4)
library(emmeans)

# time centered -> -7, 0, +7:
df <- df %>%
  mutate(time_c = time - mean(time, na.rm = TRUE))
str(df)

## systematic model comparison - global growth model
library(lme4)
library(lmerTest)
library(dplyr)

# Nullmodell: only location + time
m0_null <- lmer(log(weight) ~ location + time_c + I(time_c^2) + (1 | ID_running),
  data = df, REML = FALSE)

# Screening: all FRAC-/HRAC-groups as main effects (additive only)
m1_screen <- lmer(log(weight) ~ location + FRAC_B + FRAC_C + FRAC_D + FRAC_G +
FRAC_H + FRAC_U +
    HRAC_PDS + HRAC_PS2 + HRAC_MT + time_c + I(time_c^2) + (1 | ID_running),
  data = df, REML = FALSE)

# FRAC_C und FRAC_G + time interaction (others as main effects only)
m2_CG_time <- lmer(log(weight) ~ location +(FRAC_C + FRAC_G) * (time_c + I(time_c^2)) +
    FRAC_B + FRAC_D + FRAC_H + FRAC_U +HRAC_PDS + HRAC_PS2 + HRAC_MT + (1
| ID_running),
  data = df, REML = FALSE)

# only FRAC_C (biologically relevant group)
#     BUT no time interaction
m3_C <- lmer(log(weight) ~ location + FRAC_C + time_c + I(time_c^2) +
    (1 | ID_running),data = df, REML = FALSE)
```

```
# FRAC_C only plus effect over time?
m4_C_time <- lmer(log(weight) ~ location + FRAC_C *(time_c + I(time_c^2)) +
   (1 | ID_running),data = df, REML = FALSE)

# only C & G + time interaction
m5_CG_only_time <- lmer(log(weight) ~ location + (FRAC_C + FRAC_G) * (time_c +
I(time_c^2)) +
   (1 | ID_running), data = df, REML = FALSE)

# AIC comparison:
model_list <- list(
  m0_null        = m0_null,
  m0_null_int    = m0_null_int,
  m1_screen      = m1_screen,
  m2_CG_time     = m2_CG_time,
  m3_C           = m3_C,
  m4_C_time      = m4_C_time,
  m5_CG_only_time= m5_CG_only_time)

aic_vec <- sapply(model_list, AIC)

aic_df <- data.frame(
  Model = names(aic_vec),
  AIC   = as.numeric(aic_vec))

aic_df <- aic_df[order(aic_df$AIC), ]
aic_df$dAIC <- aic_df$AIC - min(aic_df$AIC)
print(aic_df) # 0-model best supported

summary(m0_null)
library(performance)
library(see)
check_model(m0_null)

m3_C <- update(m3_C, REML=TRUE)
check_model(m3_C)
summary(m3_C)
library(lmerTest)
anova(m3_C)

## 1.2 FRAC/HRAC final weight (31 days) - global (across locations) ++++++++++

library(readxl)
library(lme4)
library(ggplot2)
library(emmeans)
library(lmerTest)
```

```
library(dplyr)
windowsFonts(  Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial"))

df <- read_excel("data_weight_aufzuchtversuch.xlsx", sheet = "final_weight")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
str(df)
any(is.na(df))
colSums(is.na(df))

# null model: location only
m0_last_null <- lmer(weight ~ location + (1| ID_running), data = df)

# screening model: all FRAC-/HRAC-groups as main effects (additive)
m1_last_screen <- lmer(weight ~ location + FRAC_B + FRAC_C + FRAC_D + FRAC_G +
FRAC_H + FRAC_U +
    HRAC_PDS + HRAC_PS2 + HRAC_MT + (1| ID_running),  data = df)
# only FRAC_C & FRAC_G as main effects
m2_last_CG <- lmer(weight ~ location + FRAC_C + FRAC_G + (1| ID_running), data = df)
# FRAC_C  only
m3_last_C <- lmer(weight ~ location + FRAC_C + (1| ID_running), data = df)

# AIC comparison:
model_list_last <- list(
  m0_last_null     = m0_last_null,
  m1_last_screen   = m1_last_screen,
  m2_last_CG       = m2_last_CG,
  m3_last_C        = m3_last_C)

# calculate AIC
aic_vec_last <- sapply(model_list_last, AIC)

aic_last_df <- data.frame(
  Model = names(aic_vec_last),
  AIC   = as.numeric(aic_vec_last))

aic_last_df <- aic_last_df[order(aic_last_df$AIC), ]
aic_last_df$dAIC <- aic_last_df$AIC - min(aic_last_df$AIC)

print(aic_last_df) # m1_last_screen best supported

m1_last_screen <- update(m1_last_screen, REML=TRUE)
summary(m1_last_screen)
library(performance)
library(see)
```

```
check_model(m1_last_screen)
anova(m1_last_screen)

## 2.1 FRAC/HRAC indices (FRAC_load, HRAC_load, Combined_load) +++++++++++++++
#         growth models (all samplings, across locations) ++++++++++++++++++++++++++++

# - Group 1: funcidie_load -> individual FRAC Groups per sample
# - Group 2: herbicide_load -> individual HRAC Groups per sample
# - Group 3: stress_load -> combined individual FRAC/HRAC groups per sample

library(readxl)
library(lme4)
library(lmerTest)
library(dplyr)
windowsFonts(
  Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

df <- read_excel("data_weight_aufzuchtversuch.xlsx",
                 sheet = "summarized_analytics", na = "NA")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
str(df)

df <- df[!is.na(df$npest), ] # exclude all npest = NA
df <- df[!is.na(df$meanweight), ] # exclude all meanweight = NA

# time centered -> -7, 0, +7:
df <- df %>%
  mutate(time_c = time - mean(time, na.rm = TRUE))
str(df)

# create indices
df <- df %>%
  mutate(
    fungal_load    = FRAC_B + FRAC_C + FRAC_D + FRAC_G + FRAC_H + FRAC_U,
    herbicide_load = HRAC_PDS + HRAC_PS2 + HRAC_MT,
    stress_load    = fungal_load + herbicide_load)
View(df)

## systematic model comparison:
# Nullmodell (location & time only)
mA0_null <- lmer(
  log(meanweight) ~ location +
    time_c + I(time_c^2) +
    (1 + time_c | ID_running), data = df)
```

```
# fungal_load as main effect
mA1_fungal <- lmer(
  log(meanweight) ~ location +
    fungal_load +
    time_c + I(time_c^2) +
    (1 + time_c | ID_running), data = df)
# herbicide_load as main effect
mA2_herbicide <- lmer(
  log(meanweight) ~ location +
    herbicide_load +
    time_c + I(time_c^2) +
    (1 + time_c | ID_running), data = df)
# stress_load (fungicide + herbicide MoAs) as main effects
mA3_stress <- lmer(
  log(meanweight) ~ location +
    stress_load +
    time_c + I(time_c^2) +
    (1 + time_c | ID_running), data = df)
# stress_load * time (load influences growth trajectories?)
mA4_stress_time <- lmer(
  log(meanweight) ~ location +
    stress_load * (time_c + I(time_c^2)) +
    (1 + time_c | ID_running), data = df)

## AIC-comparison
model_list_A <- list(
  A0_null       = mA0_null,
  A1_fungal     = mA1_fungal,
  A2_herbicide  = mA2_herbicide,
  A3_stress     = mA3_stress,
  A4_stressTime = mA4_stress_time)

aic_A <- sapply(model_list_A, AIC)

aic_A_df <- data.frame(
  Model = names(aic_A),
  AIC   = as.numeric(aic_A))

aic_A_df <- aic_A_df[order(aic_A_df$AIC), ]
aic_A_df$dAIC <- aic_A_df$AIC - min(aic_A_df$AIC)
print(aic_A_df) # mA3_stress best model, all within delta AIC < 2

mA3_stress <- update(mA3_stress, REML=TRUE)
summary(mA3_stress)
library(performance)
library(see)
check_model(mA3_stress)
```

```
mA4_stress_time <- update(mA4_stress_time, REML=TRUE)
summary(mA4_stress_time)
confint(mA4_stress_time, parm = "stress_load:time_c")
confint(mA4_stress_time, parm = "stress_load:(time_c^2)")

ci_log <- confint(mA3_stress, parm = "stress_load")
ci_log

# Effect on original scale
beta_log <- fixef(mA3_stress)["stress_load"]
# percentage difference per additional MoA group
perc_effect <- (exp(beta_log) - 1) * 100
# transform CI
ci_perc <- (exp(ci_log) - 1) * 100
perc_effect
ci_perc

## 2.2 FRAC/HRAC indices (FRAC_load, HRAC_load, Combined_load) +++++++++++++++
#         final weight (31 days, across locations) ++++++++++++++++++++++++++++

library(readxl)
library(lme4)
library(ggplot2)
library(emmeans)
library(lmerTest)
library(dplyr)
windowsFonts(  Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial"))

df <- read_excel("data_weight_aufzuchtversuch.xlsx", sheet = "final_weight")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
str(df)
any(is.na(df))
colSums(is.na(df))

# create indices
df <- df %>%
  mutate(
    fungal_load    = FRAC_B + FRAC_C + FRAC_D + FRAC_G + FRAC_H + FRAC_U,
    herbicide_load = HRAC_PDS + HRAC_PS2 + HRAC_MT,
    stress_load    = fungal_load + herbicide_load )
any(is.na(df))
colSums(is.na(df))

## systematic model comparison:
```

```
# fungal_load
mB1_fungal <- lmer(weight ~ location + fungal_load + (1|ID_running),
  data = df, REML=FALSE)
# herbicide_load
mB2_herbicide <- lmer(weight ~ location + herbicide_load +  (1|ID_running),
  data = df, REML=FALSE)
# stress_load
mB3_stress <- lmer(weight ~ location + stress_load +  (1|ID_running),
  data = df, REML=FALSE)
# stress_load
mB4_stress_int <- lmer(weight ~ location*stress_load +  (1|ID_running),
                data = df, REML=FALSE)

nobs_vec <- c(
  B0_null = nobs(mB0_null),
  B1_fungal  = nobs(mB1_fungal),
  B2_herbicide = nobs(mB2_herbicide),
  B3_stress = nobs(mB3_stress))
print(nobs_vec) # must be equal for AIC comparison

# AIC-comparison
model_list_B <- list(
  B0_null      = mB0_null,
  B1_fungal    = mB1_fungal,
  B2_herbicide = mB2_herbicide,
  B3_stress    = mB3_stress,
  B4_stress_int = mB4_stress_int)
aic_B <- sapply(model_list_B, AIC)
aic_B_df <- data.frame(
  Model = names(aic_B),
  AIC   = as.numeric(aic_B))

aic_B_df <- aic_B_df[order(aic_B_df$AIC), ]
aic_B_df$dAIC <- aic_B_df$AIC - min(aic_B_df$AIC)

print(aic_B_df)

mB_best <- update(mB3_stress, REML = TRUE)
summary(mB_best)
confint(mB_best)
cor(df$fungal_load, df$herbicide_load)

library(performance)
library(see)
check_model(mB_best)
table(df_last$stress_load)
summary(mB3_stress)
```

```
library(performance)
library(see)
mB4_stress_int <- update(mB4_stress_int, REML = TRUE)
summary(mB4_stress_int)
check_model(mB4_stress_int)
confint(mB4_stress_int)

# -> mB3_stress best model
# -> "stress_load" effect is assumed to be linear -> for each additional MoA group XY mg     #
lower weight
# -> compare with model including “stress_load” as categorical variable
df$stress_cat <- cut(
  df$stress_load,
  breaks = c(-Inf, 0, 1, 3, Inf),
  labels = c("0", "1", "2-3", "4-6"),
  right = TRUE)
table(df$stress_cat)

# categorical model vs linear:
mB3_stress_ML <- lmer(weight ~ location + stress_load + (1|ID_running),
                      data = df, REML = FALSE)

mB4_stress_cat_ML <- lmer(weight ~ location + stress_cat + (1|ID_running),
                          data = df, REML = FALSE)

AIC(mB3_stress_ML, mB4_stress_cat_ML) # “linear” model is best supported

m_final <- update(mB3_stress_ML, REML = TRUE) # fit final selected model with REML =
TRUE

library(performance)
library(see)
summary(m_final)
check_model(m_final)
anova(m_final) # location sign.
library(emmeans)
emmeans(m_final, pairwise ~ location) # no sign. differences btw. Individual locations

# Test: does location influence pesticide effect?
boxplot(stress_load ~ location, data = df)
table(df$location, df$stress_cat)

m_main <- lmer(weight ~ location + stress_load + (1|ID_running),
               data=df, REML=FALSE)

m_int  <- lmer(weight ~ location * stress_load + (1|ID_running),
               data=df, REML=FALSE)
AIC(m_main, m_int)
```

```
# No. -> model without interaction is best supported
# --> resume with “linear” additive model!
m_final <- update(m_main, REML = TRUE)

## plotting MoA diversity effect - model-estimated ##

m_final <- lmer(weight ~ location + stress_load + (1|ID_running),
                data=df, REML=TRUE)

library(lme4)
library(ggeffects)
library(ggplot2)
library(viridis)

m_final <- update(m_main, REML = TRUE)

## Prediction data with 95% CI (fixed effects only)
## effect averaged across locations (marginal effect):

pdat <- ggpredict(m_final, terms = "stress_load [all]")

## Helper objects for plotting
stress_vals <- sort(unique(df$stress_load))

## Plot: raw data + model-based line + 95% CI ribbon
ggplot() +
  # Raw data (jittered)
  geom_jitter(
    data = df,
    aes(x = stress_load, y = weight, color = stress_load),
    width = 0.15, height = 0, size = 3, alpha = 0.8
  ) +
  # Model-based CI band
  geom_ribbon(
    data = pdat,
    aes(x = x, ymin = conf.low, ymax = conf.high),
    inherit.aes = FALSE,
    alpha = 0.2
  ) +
  # Model-based fitted line
  geom_line(
    data = pdat,
    aes(x = x, y = predicted),
    inherit.aes = FALSE,
    linewidth = 1.2,
    color = "black"
  ) +
  # Color scale for stress_load (continuous)
```

```
  scale_color_viridis_c(
    option = "plasma",
    name = "Number of MoA Groups (FRAC/HRAC)",
    breaks = stress_vals,
    direction = 1
  ) +
  # X ticks at observed stress values
  scale_x_continuous(
    breaks = stress_vals,
    minor_breaks = NULL
  ) +
  scale_y_continuous(
    limits = c(0, 180),
    breaks = seq(0, 180, by = 20)
  )+
  theme_classic(base_size = 14) +
  labs(
    x = "Number of MoA Groups in Larval Food",
    y = "Final Larval Weight (mg)",
    title = NULL
  ) +
  theme(
    legend.position   = "right",
    plot.title        = element_text(face = "bold"),
    axis.line         = element_line(color = "black", linewidth = 1),
    axis.text         = element_text(color = "black", size = 12),
    axis.ticks        = element_line(linewidth = 0.5))
# ------------------------------------------------------------
# AUTOMATED EFFECT EXTRACTION
# for final weight model - stress_load
# ------------------------------------------------------------
model <- m_final  # REML-fit
summary(m_final)
# Get range of stress_load from data file
min_MoA <- min(df$stress_load, na.rm = TRUE)
max_MoA <- max(df$stress_load, na.rm = TRUE)

# Marginal prediction (averaged across locations) at min & max:
library(emmeans)

emm <- emmeans(model, ~ stress_load,
               at = list(stress_load = c(min_MoA, max_MoA)))
emm
diff_emm <- contrast(emm, method = "revpairwise")  # max - min
diff_emm
confint(diff_emm)

# Percentage diff. -> prediction relative to min:
```

```
preds <- summary(emm)
pred_min <- preds$emmean[preds$stress_load == min_MoA]
pred_max <- preds$emmean[preds$stress_load == max_MoA]

abs_diff <- pred_max - pred_min          # (max - min)
perc_diff <- (abs_diff / pred_min) * 100  # relative to min

cat("\n----- MODEL EFFECT SUMMARY (marginal over location) -----\n")
cat("stress_load range:", min_MoA, "to", max_MoA, "\n")
cat("Predicted weight at min:", round(pred_min, 2), "mg\n")
cat("Predicted weight at max:", round(pred_max, 2), "mg\n")
cat("Absolute difference (max - min):", round(abs_diff, 2), "mg\n")
cat("Relative difference:", round(perc_diff, 1), "%\n")
# output: global across all locations

## Slope: weight diff [mg] per additional MoA-group:
model <- m_final
coefs <- coef(summary(model))
slope <- coefs["stress_load", "Estimate"]
# 95% CI for slope
ci_slope <- confint(model, parm = "stress_load")

cat("\n----- SLOPE SUMMARY -----\n")
cat("Slope (per additional MoA group):", round(slope, 3), "mg\n")
cat("95% CI:", round(ci_slope[1], 3), "to", round(ci_slope[2], 3), "mg\n")


## 2.3 FRAC/HRAC indices (Combined_load) +++++++++++++++
#         final weight (31 days, ROSENBERG) ++++++++++++++++++++++++++++

df <- read_excel("data_weight_aufzuchtversuch.xlsx", sheet = "final_weight")

df$ID_running=as.factor(df$ID_running)
df$cart=as.factor(df$cart)
df$location=as.factor(df$location)
str(df)

# Indizes erstellen
df <- df %>%
  mutate(
    fungal_load    = FRAC_B + FRAC_C + FRAC_D + FRAC_G + FRAC_H + FRAC_U,
    herbicide_load = HRAC_PDS + HRAC_PS2 + HRAC_MT,
    stress_load    = fungal_load + herbicide_load  # Gesamtbelastung)
any(is.na(df))
colSums(is.na(df))

df <- df[df$location == "Rosenberg", ] # subset Rosenberg
str(df)
```

```
View(df)
any(is.na(df))
colSums(is.na(df))

df %>%
  group_by(stress_load) %>%
  summarise(
    n_obs  = n(),
    n_cups = n_distinct(ID_running),
    .groups = "drop"
  ) %>%
  arrange(stress_load)

## model:
mB3_stress <- lmer(weight ~ stress_load +  (1|ID_running),
                   data = df, REML= TRUE)

summary(mB3_stress)
confint(mB3_stress)
library(performance)
library(see)
check_model(mB3_stress)


df$stress_cat <- cut(
  df$stress_load,
  breaks = c(-Inf, 0, 1, 3, Inf),
  labels = c("0", "1", "2-3", "4-6"),
  right = TRUE)
table(df$stress_cat)

# categorical model vs linear:
mB3_stress_ML <- lmer(weight ~ stress_load + (1|ID_running),
                      data = df, REML = FALSE)

mB4_stress_cat_ML <- lmer(weight ~ stress_cat + (1|ID_running),
                          data = df, REML = FALSE)

AIC(mB3_stress_ML, mB4_stress_cat_ML) # linear model is best supported

mB3_stress <- update(mB3_stress, REML = TRUE)

## plot stress load effect day 31 ROSENBERG:

library(lme4)
library(ggeffects)
library(ggplot2)
library(viridis)
```

```
m_final <- mB3_stress

##  Create prediction data with 95% CI (fixed effects only)
## effect averaged across locations (marginal effect):
pdat <- ggpredict(m_final, terms = "stress_load [all]")

## Helper objects for plotting
stress_vals <- sort(unique(df$stress_load))

##  Plot: raw data + model-based line + 95% CI ribbon
ggplot() +
  geom_jitter(
    data = df,
    aes(x = stress_load, y = weight, color = stress_load),
    width = 0.15, height = 0, size = 3, alpha = 0.8
  ) +
    geom_ribbon(
    data = pdat,
    aes(x = x, ymin = conf.low, ymax = conf.high),
    inherit.aes = FALSE,
    alpha = 0.2
  ) +
    geom_line(
    data = pdat,
    aes(x = x, y = predicted),
    inherit.aes = FALSE,
    linewidth = 1.2,
    color = "black"
  ) +
  scale_color_viridis_c(
    option = "plasma",
    name = "Number of MoA Groups (FRAC/HRAC)",
    breaks = stress_vals,
    direction = 1
  ) +
  scale_x_continuous(
    breaks = stress_vals,
    minor_breaks = NULL
  ) +
  scale_y_continuous(
    limits = c(0, 180),
    breaks = seq(0, 180, by = 20)
  )+
  theme_classic(base_size = 14) +
  labs(
    x = "Number of MoA Groups in Larval Food",
    y = "Final Larval Weight after 31 days [mg]",
```

```
    title = NULL
  ) +
  theme(
    legend.position   = "right",
    plot.title        = element_text(face = "bold"),
    axis.line         = element_line(color = "black", linewidth = 1),
    axis.text         = element_text(color = "black", size = 12),
    axis.ticks        = element_line(linewidth = 0.5))

# model estimates:
library(emmeans)
model <- m_final  # REML-fit

min_MoA <- min(df$stress_load, na.rm = TRUE)
max_MoA <- max(df$stress_load, na.rm = TRUE)

# Marginal predictions at min & max (averaged over location)
emm <- emmeans(model, ~ stress_load,
               at = list(stress_load = c(min_MoA, max_MoA)))

# Predicted means + 95% CI for min & max
emm_ci <- confint(emm)
emm_ci
# Contrast (max - min) + 95% CI
diff_emm <- contrast(emm, method = "revpairwise")  # max - min
diff_ci  <- confint(diff_emm)
# extract values
preds <- as.data.frame(emm_ci)

pred_min <- preds$emmean[preds$stress_load == min_MoA]
pred_max <- preds$emmean[preds$stress_load == max_MoA]

ci_min <- preds[preds$stress_load == min_MoA, c("lower.CL", "upper.CL")]
ci_max <- preds[preds$stress_load == max_MoA, c("lower.CL", "upper.CL")]

abs_diff  <- pred_max - pred_min
perc_diff <- (abs_diff / pred_min) * 100

cat("\n----- MODEL EFFECT SUMMARY (marginal over location) -----\n")
cat("stress_load range:", min_MoA, "to", max_MoA, "\n")
cat("Predicted weight at min:", round(pred_min, 2), "mg",
    "(95% CI:", round(ci_min$lower.CL, 2), "to", round(ci_min$upper.CL, 2), ")\n")
cat("Predicted weight at max:", round(pred_max, 2), "mg",
    "(95% CI:", round(ci_max$lower.CL, 2), "to", round(ci_max$upper.CL, 2), ")\n")
cat("Absolute difference (max - min):", round(abs_diff, 2), "mg\n")
print(diff_ci)
cat("Relative difference:", round(perc_diff, 1), "%\n")
```

```
                ##########################################
++++++++++++++##   Contaminated Host Plants – larval SURVIVAL   ##++++++++++++++
                ##########################################

library(readxl)
library(dplyr)

df <- read_excel("...link to datafile...",  "summarized_analytics")
any(is.na(df))
colSums(is.na(df))

df_last <- df %>% filter(time == 31) # subset for final weight (time = 31 days)
str(df_last)

df_last$ID_running=as.factor(df_last$ID_running)
df_last$cart=as.factor(df_last$cart)
df_last$location=as.factor(df_last$location)
df_last$sumconpest=as.numeric(df_last$sumconpest)
df_last$npest=as.numeric(df_last$npest)
str(df_last)

## preparation of variables

# npest as categorical:
df_last$npest_cat <- cut(
  df_last$npest,
  breaks = c(-Inf, 0, 1, 4, 7, Inf),
  labels = c("0", "1", "2-4", "5-7", "8-9"),
  right = TRUE)

df_last$npest_cat <- droplevels(df_last$npest_cat)
table(df_last$npest_cat)

# Create MoA indices (based on FRAC/HRAC groups):
frac_cols <- c("FRAC_B", "FRAC_C", "FRAC_D", "FRAC_G", "FRAC_H", "FRAC_U")
hrac_cols <- c("HRAC_PDS", "HRAC_PS2", "HRAC_MT")

df_last$fungicide_load <- rowSums(df_last[, frac_cols], na.rm = TRUE) # Number of FRAC-
groups per plant ID
df_last$herbicide_load <- rowSums(df_last[, hrac_cols], na.rm = TRUE) # Number of HRAC-
groups per plant ID
df_last$combined_load <- df_last$fungicide_load + df_last$herbicide_load # Total Number of
distinct MoA-groups

summary(df_last$fungicide_load)
table(df_last$fungicide_load)
summary(df_last$herbicide_load)
```

```
table(df_last$herbicide_load)
summary(df_last$combined_load)
table(df_last$combined_load)

## SURVIVAL ANALYSIS at time == 31 –> model screening

# Check some models for overdispersion -> common in binomial survival models, if present, AIC comparison not viable!

mS0_null <- glm(cbind(alive, total - alive) ~ location, data = df_last, family = binomial)
mS1_sumcon <- glm(cbind(alive, total - alive) ~ location + sumconpest, data = df_last, family = binomial)
mS2_npest_num <- glm(cbind(alive, total - alive) ~ location + npest, data = df_last, family = binomial)
mS3_npest_cat <- glm(cbind(alive, total - alive) ~ location + npest_cat, data = df_last, family = binomial)
mS4_fungicide <- glm(cbind(alive, total - alive) ~ location + fungicide_load, data  = df_last, family = binomial)
mS5_herbicide <- glm(cbind(alive, total - alive) ~ location + herbicide_load, data = df_last, family = binomial)
mS6_combined <- glm(cbind(alive, total - alive) ~ location + combined_load, data = df_last, family = binomial)

models <- list(
  mS0_null      = mS0_null,
  mS1_sumcon    = mS1_sumcon,
  mS2_npest_num = mS2_npest_num,
  mS3_npest_cat = mS3_npest_cat,
  mS4_fungicide = mS4_fungicide,
  mS5_herbicide = mS5_herbicide,
  mS6_combined  = mS6_combined)

phi_values <- sapply(models, function(m) deviance(m) / df.residual(m))
phi_values # no problematic overdispersion -> range phi (1.18-1.2) -> AIC comparion viable

range(fitted(mS0_null)) # no extreme values e.g. near '0' or '1' -> binomial model is stable

## proceed with AIC model screening:

# all candidate models

# null-model:
mS0_null <- glm(cbind(alive, total - alive) ~ location, data = df_last, family = binomial)
# sum concentration (sumconpest, continuous):
mS1_sumcon <- glm(cbind(alive, total - alive) ~ location + sumconpest, data = df_last, family = binomial)
# number of pesticides – numeric:
```

```
mS2_npest_num <- glm(cbind(alive, total - alive) ~ location + npest, data = df_last, family = binomial)
# number of pesticides – categorical (0, 1, 2–4, 5–7, 8–9):
mS3_npest_cat <- glm(cbind(alive, total - alive) ~ location + npest_cat, data = df_last, family = binomial)
# fungicide MoA diversity (number of FRAC-groups):
mS4_fungicide <- glm(cbind(alive, total - alive) ~ location + fungicide_load, data  = df_last, family = binomial)
# herbicide MoA diversity (number of HRAC-groups):
mS5_herbicide <- glm(cbind(alive, total - alive) ~ location + herbicide_load, data = df_last, family = binomial)
# combined MoA diversity (individual FRAC + HRAC-groups):
mS6_combined <- glm(cbind(alive, total - alive) ~ location + combined_load, data = df_last, family = binomial)
# individual MoA groups independently:
mS7_all_MoA <- glm(cbind(alive, total - alive) ~ location + FRAC_B + FRAC_C + FRAC_D + FRAC_G + FRAC_H + FRAC_U + HRAC_PDS + HRAC_PS2 + HRAC_MT, data  = df_last, family = binomial)
# only FRAC_C and FRAC_G:
mS8_CG <- glm(cbind(alive, total - alive) ~ location + FRAC_C + FRAC_G, data  = df_last, family = binomial)
# interaction FRAC_C * FRAC_G (combined effect):
mS9_CG_int <- glm(cbind(alive, total - alive) ~ location + FRAC_C * FRAC_G, data  = df_last, family = binomial)

# AIC-comparison of all survival-models

model_list_surv <- list(
  S0_null     = mS0_null,
  S1_sumcon   = mS1_sumcon,
  S2_npest_num = mS2_npest_num,
  S3_npest_cat = mS3_npest_cat,
  S4_fungicide = mS4_fungicide,
  S5_herbicide = mS5_herbicide,
  S6_combined  = mS6_combined,
  S7_all      = mS7_all_MoA,
  S8_CG       = mS8_CG,
  S9_CGint    = mS9_CG_int)

aic_surv_vec <- sapply(model_list_surv, AIC)

aic_surv_df <- data.frame(
  Model = names(aic_surv_vec),
  AIC   = as.numeric(aic_surv_vec))

# sort &  calculate ΔAIC
aic_surv_df <- aic_surv_df[order(aic_surv_df$AIC), ]
aic_surv_df$dAIC <- aic_surv_df$AIC - min(aic_surv_df$AIC)
```

```
print(aic_surv_df) # sumconpest -> best model

summary(mS1_sumcon) # sumconpest n.s.
library(DHARMa)
res_sumcon <- simulateResiduals(mS1_sumcon, n = 1000)
plot(res_sumcon)
testDispersion(res_sumcon)
# model fits well

##  % survival rate – global:

# survival proportion per plant ID
df_last$surv_prop <- df_last$alive / df_last$total
# overall survival proportion
mean_surv <- mean(df_last$surv_prop, na.rm = TRUE)
mean_surv
# larval survival (global): 81.3%
                    ########################################
 ++++++++++++++++##   Helicopter Spray Exposure – larval weight   ##++++++++++++++++
                    ########################################

library(readxl)
data<- read_excel("Auswertungstabelle_Gewicht.xlsx", sheet = "weight_spritzversuch")

df <- na.omit(data) # exclude data with missing values

df$ID=as.factor(df$ID)
df$cart=as.factor(df$cart)
df$cohort=as.factor(df$cohort) #cohort = caterpillar background (pre-trial vs. control rearing)
df$treatment=as.factor(df$treatment)
df$weight=as.numeric(df$weight)
str(df)
View(df)

table(df$ID, df$time) # most obsevations are from T1, only 46 have an additional T2 value

# day 11: n=133 observations
# day 20: n=46 observations
# day 29: n=8 observations ->  exclude from analysis

## Approach:
# - Unbalanced repeated measures (observations: 133 IDs T1, only 46 IDs T1+T2)
# -> no growth model
#
# Analysis (all data, T1 + T2):
# - includes all observations
# - differences in mean weight over time (BUT no individual growth inference!)
```

```
#
# 'start_weight' as baseline co-variate
# 'time' centered on T2 for interpretation of late effects
#         -> expectation: if present, effects will be detected at later stages
# 'cohort' as co-variate for larval origin (pre-experiment vs. control rearing)

## data prep:

library(dplyr)
df2 <- df %>%
  mutate(
    ID = factor(ID),
    time = as.numeric(time)
  ) %>%
  # keep only days 11 & 20 (drop 29)
  filter(time %in% c(11, 20)) %>%

# time centered on day 20 -> Intercept = expected weight at day 20, not at T0
  mutate(time_c = time - 20)

# cohorts -> start_weight mean values (+SE):
df_all %>%
  group_by(cohort) %>%
  summarise(
    n = n(),
    mean_start_weight = mean(start_weight, na.rm = TRUE),
    sd_start_weight   = sd(start_weight, na.rm = TRUE))

df_all <- df2

# Baseline weight log-transformed
df_all <- df_all %>%
  mutate(log_start_weight = log10(start_weight))

## linear model:
m1 <- lm(log_weight ~ treatment * time_c + log_start_weight + cohort + cart,
         data = df_all)

summary(m1)
library(performance)
check_model(m1)
library(car)

Anova(m1, type = 2)

# treatment n.s.
# time & start_weight significant
# cohort significant!
```

```
## test model including treatment*cohort inteaction:
m2 <- lm(log_weight ~ treatment * time_c + treatment * cohort +
    log_start_weight + cart, data = df_all)

summary(m2)
check_model(m2)
Anova(m2, type = 2) # -> treatment response was not influenced by larval cohort


                    ########################################
  ++++++++++++++##   Helicopter Spray Exposure – larval survival   ##++++++++++++++++
                    ########################################


library(readxl)
library(lme4)
library(ggplot2)
windowsFonts(Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

df<- read_excel("Auswertungstabelle_Alive_Dead.xlsx", sheet = "survival_caterpillars")

any(is.na(df))
df$cart=as.factor(df$cart)
df$cohort=as.factor(df$cohort) #cohort = caterpillar background (pre-trial vs. control rearing)
df$location=as.factor(df$location)
df$treatment=as.factor(df$treatment)
df$start_weight=as.numeric(df$start_weight)
str(df)
View(df)

## glm()  binary response (link.logit)

# base model without interactions:
m_base <- glm(
  survival ~ treatment + start_weight + cohort,
  family = binomial,
  data = df) # “+ cart” excluded based on AIC comp. and anova model-comp.

# model including interaction:
m_int <- glm(
  survival ~ treatment * cohort + start_weight,
  family = binomial,
  data = df)

anova(m_base, m_int, test = "Chisq") # m_int is significantly better
phi <- (deviance(m_int) / df.residual(m_int))
phi #  phi = 0.55
```

```
range(fitted(m_int))

# proceed with m_int:
m1 <- m_int

library(DHARMa)
res <- simulateResiduals(m1)
plot(res)
testDispersion(res)
testZeroInflation(res)

summary(m1) # separation? interactions show low estimates & part. high SEs!
with(df, table(survival, treatment, cohort)) # check for separation
# -> no dead larvae in the control in the cohort "control_rearing"

# ignore wald-test from model summary:
library(emmeans)
emm_surv <- emmeans(
  m1, ~ treatment | cohort,
  type = "response")
emm_surv # survival rates per treatment & cohort

pairs(emm_surv, adjust = "tukey") # pairwise comparisons
pairs(emm_surv)


## stacked bar chart for survival vs. treatment:
library(dplyr)
library(ggplot2)
library(scales)

# Prep: define factor levels
df2 <- df %>%
  mutate(
    survival_f = factor(ifelse(survival == 1, "alive", "dead"),
                        levels = c("dead", "alive")),
    cohort     = factor(cohort, levels = c("control_rearing", "pre_trial")),
    treatment  = factor(treatment, levels = c("K", "drift_rate", "field_rate")))

# Aggregate per treatment x cohort x survival
df_counts <- df2 %>%
  count(treatment, cohort, survival_f, name = "count") %>%
  group_by(treatment, cohort) %>%
  mutate(
    fraction = count / sum(count),
    ymax = cumsum(fraction),
    ymin = ymax - fraction,
    ytext = (ymin + ymax) / 2
```

```
  ) %>%
  ungroup()

# x-axis order -> two bars per treatment (cohort next to each other), then spacer
x_levels <- c(
  "K\ncontrol_rearing", "K\npre_trial", " ",            # Spacer
  "drift_rate\ncontrol_rearing", "drift_rate\npre_trial", "  ",
  "field_rate\ncontrol_rearing", "field_rate\npre_trial")

df_counts <- df_counts %>%
  mutate(
    x_tc = paste0(as.character(treatment), "\n", as.character(cohort)),
    x_tc = factor(x_tc, levels = x_levels))

# dummy row for spacer -> invisible bar to separate treatment bars
df_spacer <- tibble(
  x_tc = factor(c(" ", "  "), levels = x_levels),
  fraction = NA_real_,
  survival_f = factor("dead", levels = c("dead", "alive")),
  count = NA_integer_,
  ytext = NA_real_)

df_plot <- bind_rows(df_counts, df_spacer)

# Plot:
p_surv <- ggplot(df_plot, aes(x = x_tc, y = fraction, fill = survival_f)) +
  geom_col(width = 0.5, color = "black", linewidth = 0.8, na.rm = TRUE) +
  geom_text(aes(y = ytext, label = count),
            color = "black",
            size = 4.2,
            fontface = "plain",
            na.rm = TRUE) +
  scale_y_continuous(
    labels = percent,
    limits = c(-0.06, 1),
    expand = expansion(mult = c(0, 0.05))
  ) +
  scale_x_discrete(
    labels = c(
      "K\ncontrol_rearing"         = "Control\nControl Reared",
      "K\npre_trial"               = "Control\nPre-Trial",
      " "                          = "",
      "drift_rate\ncontrol_rearing" = "Drift rate\nControl Reared",
      "drift_rate\npre_trial"       = "Drift rate\nPre-Trial",
      "  "                         = "",
      "field_rate\ncontrol_rearing" = "Field rate\nControl Reared",
      "field_rate\npre_trial"       = "Field rate\nPre-Trial"
    )
```

```
  ) +
  scale_fill_manual(
    values = c(dead = "grey60", alive = "seagreen3"),
    name = NULL   # keine Legendenüberschrift
  ) +
  guides(
    fill = guide_legend(
      override.aes = list(color = "black", linewidth = 0.8)
    )
  ) +
  labs(x = NULL, y = "Survival (%)") +
  theme_minimal(base_size = 15) +
  theme(
    panel.grid = element_blank(),
    axis.line = element_line(color = "black", linewidth = 0.8),
    axis.text = element_text(color = "black", size = 13),
    axis.title = element_text(size = 14),
    legend.position = "right",
    legend.key.width  = unit(0.6, "cm"),
    legend.key.height = unit(0.6, "cm")
  )
p_surv


                ################################################
++++++++++++##    Luna® Experience - ISOLATED larvae ->  survival     ##++++++++++++
                ################################################

library(readxl)
library(ggplot2)
windowsFonts(
  Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial"))

data<- read_excel("...link to datafile...", sheet = "survival")

any(is.na(df))
colSums(is.na(df))
df$treatment=as.factor(df$treatment)
str(df)
View(df)

# glm() with binary(link.logit)

m1 <- glm(survival ~ treatment + start_weight,
          family = binomial(link = "logit"),
          data = df)
phi <- (deviance(m1) / df.residual(m1))
phi #  phi = 0.97
```

```
range(fitted(m1))

exp(coef(m1)) # mutate coefficients from logit-scale (log-Odds) to odds-ratios

summary(m1)
library(DHARMa)
sim_res <- simulateResiduals(m1)
plot(sim_res)

library(car)
Anova(m1, type = "II", test.statistic = "LR")

# estimated marginal means (on logit-scale)
library(emmeans)
emm <- emmeans(m1, ~ treatment)

# group comparisons (post-hoc contrast)
contr <- pairs(emm)
library(ggplot2)
library(dplyr)
#  extract data and calculate OR + CI
df_or <- summary(contr) %>%
  as.data.frame() %>%
  mutate(
    # OR from logit-difference
    OR = exp(estimate),
    # 95% CI to logit-scale (transformation)
    lower.CI = exp(estimate - 1.96 * SE),
    upper.CI = exp(estimate + 1.96 * SE),
    # labelled CIs
    CI = paste0(round(lower.CI,2), " – ", round(upper.CI,2))
  ) %>%
  select(contrast, OR, CI, z.ratio, p.value)
df_or

# survival probabilities:
emmeans(m1, ~ treatment, type = "response")

++++++++++++++++++++++++++++++++++++++++++++++++++

# stacked bar chart: survival vs. treatment
library(ggplot2)
library(dplyr)

# rename & order Treatment levels
df <- df %>%
  mutate(survival = factor(survival, levels = c(0, 1), labels = c("Dead", "Alive")))
```

```
# calculate number of obs. survival per level and treatment
df_counts <- df %>%
  group_by(treatment, survival) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  group_by(treatment) %>%
  mutate(
    fraction = count / sum(count),
    ymax = cumsum(fraction),
    ymin = ymax - fraction,
    ytext = (ymin + ymax)/2)

# stacked bars plot
plot <- ggplot(df_counts, aes(x = treatment, y = fraction, fill = survival)) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_text(aes(y = ytext, label = count), color = "white", size = 5) +
  scale_y_continuous(labels = scales::percent, expand = expansion(mult = c(0, 0.05))) +
  scale_fill_manual(values = c("grey60", "seagreen3")) +
  labs(
    x = "Treatment",
    y = "Proportion of Overall Survival",
    fill = "Survival",
    title = NULL
  ) +
  theme_minimal(base_size = 15) +
  theme(
    panel.grid = element_blank(),
    axis.line = element_line(color = "black"),
    plot.title = element_text(hjust = 0.5))
plot

              ######################################################
++++++++++##    Luna® Experience - ISOLATED larvae ->  larval growth    ##++++++++++
              ######################################################

library(readxl)
library(lme4)
library(ggplot2)
library(lmerTest)
library(dplyr)

windowsFonts(
  Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

data <- read_excel("...link to datafile...", sheet = "weight")

df$ID=as.factor(df$ID)
```

```
df$sampling=as.factor(df$sampling)
df$treatment=as.factor(df$treatment)
df$weight=as.numeric(df$weight)
df$start_weight=as.numeric(df$start_weight)
df$time=as.numeric(df$time)
str(df)
View(df)

## growth model (all samplings):

# time centered -> -4.5, 1.5, +7.5
df <- df %>%
  mutate(time_c = time - mean(time, na.rm = TRUE))
str(df)
table(df$time_c)

# model
contrasts(df$treatment) <- contr.sum(2) # set contrasts to test treatment differences for mean
growth trajectory, not at reference time=0
contrasts(df$treatment)

m_growth <- lmer(
  log(weight) ~ treatment * (time_c + I(time_c^2)) + start_weight+
    (1 + time_c | ID),
  data = df)

summary(m_growth)
library(performance)
library(see)
check_model(m_growth)
library(lmerTest)
anova(m_growth, type=3)

++++++++++++++++++++++++++++++++++++++++++++++++++++++++

# plotting growth trajectories per treatment on original scales (time (days), weight (mg))

library(dplyr)
library(ggplot2)
library(viridis)

time_center <- mean(df$time, na.rm = TRUE)

time_seq <- seq(
  from = min(df$time, na.rm = TRUE),
  to   = max(df$time, na.rm = TRUE),
  length.out = 100)
```

```
newdat <- expand.grid(
  time      = time_seq,
  treatment = levels(df$treatment))

newdat <- newdat %>%
  mutate(
    time_c       = time - time_center,   # exakt wie im Modell
    `I(time_c^2)` = time_c^2,
    start_weight  = mean(df$start_weight, na.rm = TRUE))

pred <- predict(
  m_growth, newdata = newdat,
  re.form = NA, se.fit = TRUE)

newdat$log_weight_hat <- pred$fit
newdat$se_log         <- pred$se.fit

alpha <- 0.05
z_crit <- qnorm(1 - alpha/2)   # ~1.96

newdat <- newdat %>%
  mutate(
    log_low   = log_weight_hat - z_crit * se_log,
    log_high  = log_weight_hat + z_crit * se_log,
    weight_hat = exp(log_weight_hat),
    weight_low = exp(log_low),
    weight_high = exp(log_high))

ggplot() +
  # raw data jitter
  geom_jitter(
    data = df,
    aes(x = time, y = weight, colour = treatment),
    width  = 0.2,
    height = 0,
    alpha  = 0.5,
    size   = 1.7
  ) +
    # model curve
  geom_line(
    data = newdat,
    aes(x = time, y = weight_hat, colour = treatment),
    linewidth = 1.4
  ) +
  # dashed CI-boreders
  geom_line(
    data = newdat,
    aes(x = time, y = weight_low, colour = treatment),
```

```
    linetype = "dashed",
    linewidth = 0.8
  ) +
  geom_line(
    data = newdat,
    aes(x = time, y = weight_high, colour = treatment),
    linetype = "dashed",
    linewidth = 0.8
  ) +
  scale_color_manual(
    values = c(
      "Control" = "black",
      "LUNA"    = "steelblue2"
    )) +
  scale_x_continuous(breaks = c(7, 13, 19)) +
  scale_y_log10(
    breaks = c(30, 50, 100, 200, 400, 800, 1600),
    labels = c(30, 50, 100, 200, 400, 800, 1600)
  ) +
  coord_cartesian(ylim = c(25, 1600))+
  labs(
    x = "Time (days)",
    y = "Larval Weight [mg]",
    colour = "Treatment"
  ) +
  theme_classic(base_size = 16) +
  theme(
    legend.position = "right",
    axis.text  = element_text(size = 14, colour = "black"),
    axis.title = element_text(size = 16, face = "bold"),
    axis.line = element_line(linewidth = 1, colour = "black"),
    panel.border = element_blank(),
    legend.title = element_text(size = 14, face = "bold"),
    legend.text  = element_text(size = 12))
# plot shows growth trajectories per treatment
# weight is back transformed to original scale, y-axis is on log10

            ##########################################################
 ++++++++##    Luna® Experience - ISOLATED larvae ->  final larval weight    ##+++++++++
            ##########################################################

library(readxl)
library(lme4)
library(ggplot2)
library(dplyr)

windowsFonts(
  Arial = windowsFont("Arial"))
```

```
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

data <- read_excel("...link to datafile...", sheet = "weight")

df$ID=as.factor(df$ID)
df$sampling=as.factor(df$sampling)
df$treatment=as.factor(df$treatment)
df$weight=as.numeric(df$weight)
df$start_weight=as.numeric(df$start_weight)
df$time=as.numeric(df$time)
str(df)
View(df)

# subset for time = "19" (days)
df_last <- df %>%
  filter(time == 19)
View(df_last)

## center start_weight:
df_last <- df_last %>%
  mutate(start_c = start_weight - mean(start_weight, na.rm = TRUE))

# Centering start_weight enables the interpretation of the
# main treatment effects and makes the treatment:start_weight interaction
# numerically stable

## models -> basic vs interaction model (does start_weight influence treatment effect?)

m_last <- lm(weight ~ treatment + start_c, data = df_last)
summary(m_last)

m_last_int <- lm(weight ~ treatment * start_c, data = df_last)
summary(m_last_int)

anova(m_last, m_last_int) # interaction not needed -> m_last best model

library(performance)
library(see)
check_model(m_last)

library(car)
Anova(m_last, type = 2) # treatment n.s.

library(emmeans)
emm_last <- emmeans(m_last, ~ treatment) # Estimated marginal means (adjusted)
emm_last

                ##################################################
```

```
++++++++++##    Luna® Experience - ISOLATED larvae ->  adult weight    ##++++++++++++
              ##################################################

library(readxl)
library(ggplot2)
windowsFonts(
  Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

data_adults <- read_excel("...link to datafile...", sheet = "imagines")

dfa <- na.omit(data_adults)

dfa$ID=as.factor(dfa$ID)
dfa$treatment=as.factor(dfa$treatment)
dfa$weight=as.numeric(dfa$weight)

str(dfa)
View(dfa)

# subsets male only (no females in LUNA Group)
df_a1 <- subset(dfa, sex == "m") # 16 observations -> LUNA 4 / Control 12
View(df_a1)

## test for data distribution:
# LUNA: n=4 -> shapiro alone not reliable for low n -> visual inspection with histograms

by(df_a1$weight, df_a1$treatment, shapiro.test) # shapiro:
# histrogram:
par(mfrow = c(1, 2))  # two plots next to each other

with(subset(df_a1, treatment == "Control"),
     hist(weight, main = "Control", col = "grey", xlab = "Weight [mg]"))

with(subset(df_a1, treatment == "LUNA"),
     hist(weight, main = "LUNA", col = "lightblue", xlab = "Weight [mg]"))

par(mfrow = c(1, 1))

# QQ-plot:
par(mfrow = c(1, 2))
with(subset(df_a1, treatment == "Control"), {
  qqnorm(weight, main = "QQ-Plot Control")
  qqline(weight)
})
with(subset(df_a1, treatment == "LUNA"), {
  qqnorm(weight, main = "QQ-Plot LUNA")
  qqline(weight)
```

```
})
par(mfrow = c(1, 1))

# wilcox:
wilcox.test(weight ~ treatment, data = df_a1) # n.s.
#t.test(weight ~ treatment, data = df_a1, var.equal = FALSE) # same result


                  ################################################
++++++++++++##    Luna® Experience - GROUPED larvae ->  survival    ##++++++++++++
                  ################################################

library(readxl)
library(lme4)
library(ggplot2)
library(emmeans)
library(dplyr)
windowsFonts(
  Arial = windowsFont("Arial")
)
theme_set(theme_classic(base_size = 14, base_family = "Arial")) # set font for all ggplots

df <- read_excel("...link to datasheet...",
                 sheet = "survival")
any(is.na(df))
colSums(is.na(df))

df$treatment=as.factor(df$treatment)
df$start_state=as.factor(df$start_state) # L4 or L5
df$ID=as.factor(df$ID)

str(df)
View(df)

# start state 'L3' excluded from table (i.e. analysis) due to 100% mortality across treatments

# Fishers Exact Test -> due to low number of biological replicates (i.e. housing boxes) of 5 per
treatment (1x L5, 4x L4)

library(dplyr)

# Summarize Dataframe:
df_summary <- df %>%
  group_by(treatment) %>%
  summarise(
    total_insects = sum(total),       # total number of adults
    total_alive = sum(alive),         # number adults alive
    total_dead = sum(dead),           # number adults dead
    survival = total_alive / total_insects,  # survival rate
```

```
    survival_se = sqrt(survival * (1 - survival) / total_insects))  # SE of survival rate
df_summary

# Contingency Table:
cont_table <- matrix(c(
  df_summary$total_dead[df_summary$treatment == "Kontrolle"],  # dead in control
  df_summary$total_alive[df_summary$treatment == "Kontrolle"],  # alive in control
  df_summary$total_dead[df_summary$treatment == "LUNA"],       # dead in Luna
  df_summary$total_alive[df_summary$treatment == "LUNA"]       # alive in Luna
), nrow = 2, byrow = TRUE,
dimnames = list(Treatment = c("Kontrolle", "LUNA"), Outcome = c("Dead", "Alive")))  # name
rows and lines

# Fisher's Exact Test:
fisher_test_result <- fisher.test(cont_table)
print(fisher_test_result)

++++++++++++++++++++++++++++++++++++++++++++++++++

## stacked bar plot

data<- read_excel("...link to datasheet...", sheet = "survival")
df <- na.omit(data)
df$treatment=as.factor(df$treatment)

# rename & order treatment levels
df <- df %>%
  mutate(survival = factor(survival, levels = c(0, 1), labels = c("Dead", "Alive")))

# calculate number of obs. survival per level and treatment
df_counts <- df %>%
  group_by(treatment, survival) %>%
  summarise(count = n()) %>%
  ungroup() %>%
  group_by(treatment) %>%
  mutate(
    fraction = count / sum(count),
    ymax = cumsum(fraction),
    ymin = ymax - fraction,
    ytext = (ymin + ymax)/2)

# stacked bars plot
plot <- ggplot(df_counts, aes(x = treatment, y = fraction, fill = survival)) +
  geom_bar(stat = "identity", width = 0.7, color = "black") +
  geom_text(aes(y = ytext, label = count), color = "black", size = 5) +
  scale_y_continuous(
    labels = scales::percent,
    limits = c(-0.05, 1),        # show 5% below zero
```

```
    expand = expansion(mult = c(0, 0.05))
  ) +
  scale_fill_manual(values = c("grey60", "seagreen3")) +
  labs(
    x = "Treatment",
    y = "Proportion of Overall Survival",
    fill = "Survival",
    title = NULL
  ) +
  theme_minimal(base_size = 15) +
  theme(
    panel.grid = element_blank(),
    axis.line = element_line(color = "black"),
    plot.title = element_text(hjust = 0.5),
    axis.text = element_text(color = "black"),
    legend.position = "none")
plot

#######################################################
##    Luna® Experience - GROUPED larvae ->  adult weight    ##
#######################################################

library(readxl)
library(lme4)
library(lmerTest)
library(ggplot2)
windowsFonts(Arial = windowsFont("Arial"))
theme_set(theme_classic(base_size = 14, base_family = "Arial"))

df <- read_excel("...link to datasheet...")
any(is.na(df))
colSums(is.na(df))
df$treatment <- factor(df$treatment)
df$start_state <-factor(df$start_state) # L4 or L5
df$sex <- factor(df$sex)
df$group <- factor(df$group)

str(df)
View(df)

# LMM accounting for sex & start_state (L4/L5):

m1 <- lmer(weight ~ treatment + sex + start_state + (1|group), data = df)

summary(m1)
library(performance)
library(see)
check_model(m1)
```

```
library(lmerTest)
anova(m1, type = 2)
```