Ministerium für Umwelt, Klima und Energiewirtschaft Baden-Württemberg

Improving the Reliability of Decision-Support Systems for Nuclear Emergency Management Using Software Diversity Methods

Tudor B. Ionescu, Eckart Laurien, Walter Scheuermann
tudor.ionescu@ike.uni-stuttgart.de

Institute of Nuclear Technology and Energy Systems
Stuttgart, Germany

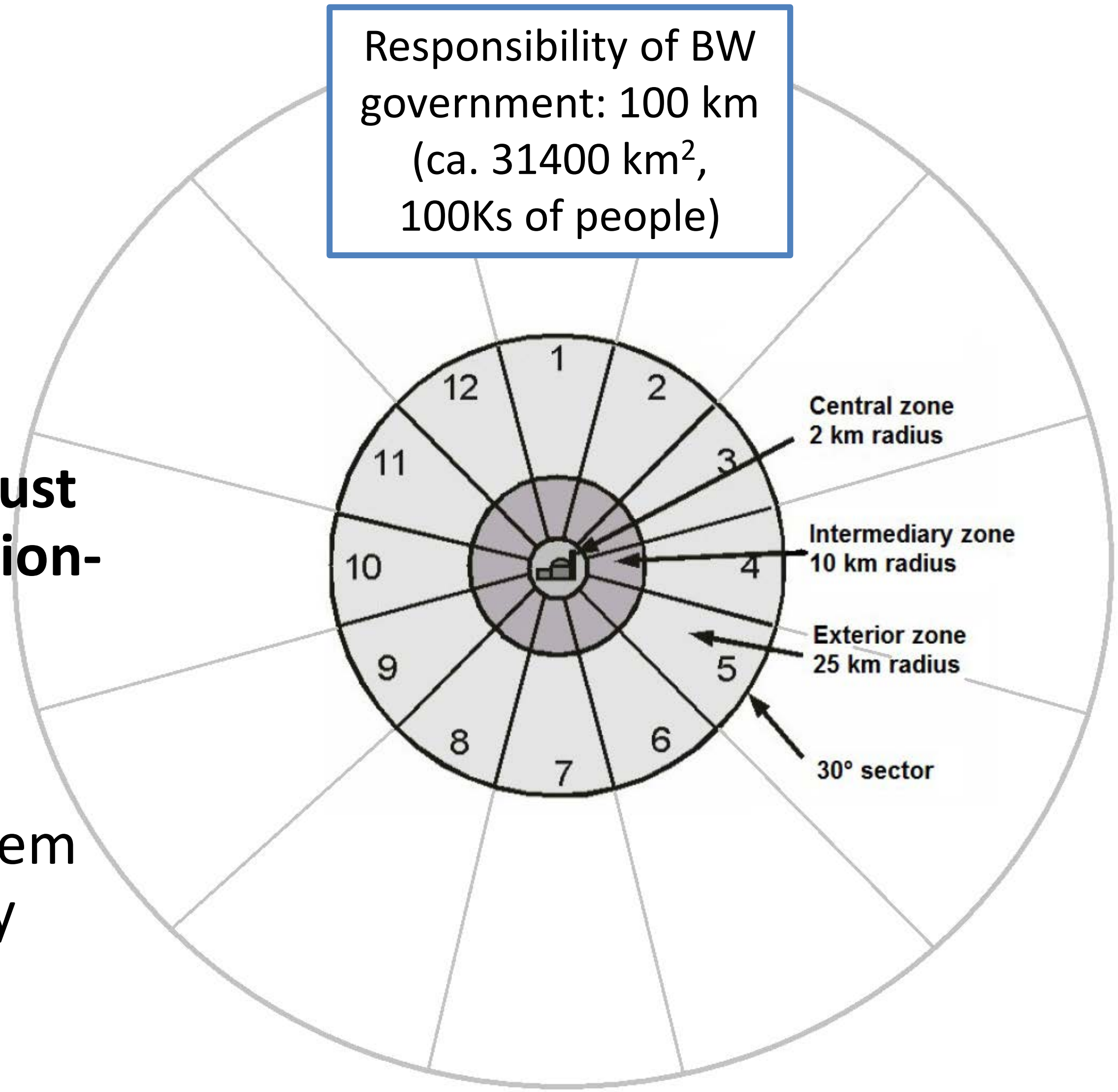
16th International Conference on Harmonisation within Atmospheric Dispersion Modelling for Regulatory Purposes
8-11 September 2014, Varna, Bulgaria

Taking Counter-Measures in Nuclear Emergencies

- Resources are limited
- Time is critical
- Emission data is scarce
- ... (unknown factors)

- **Can decision-makers trust the results of a simulation-based DSNE system in emergency situations?**

- Fukushima: SPEEDI system was not used effectively
  - Source: Japanese Diet's Report on Fukushima

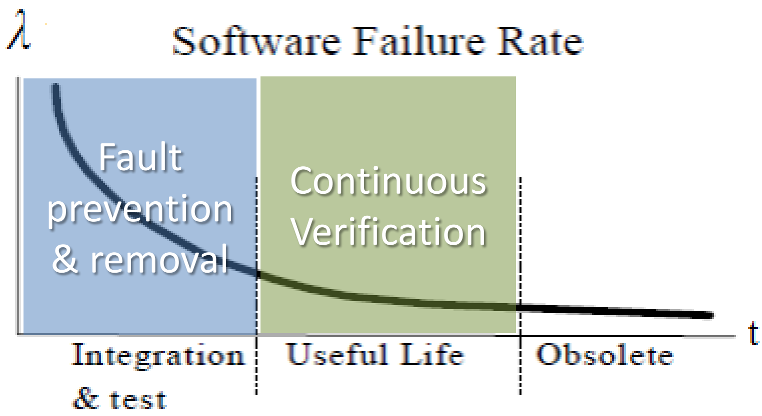


Motivation

- NASA-STD-8719.13B standard for safety-critical software, paragraph 4.1.1.2.a:
  - As soon as the software *"processes data or analyzes trends that lead directly to safety decisions"* it must be considered safety critical → **DSNE systems are safety-critical**
- From a Software Engineering point of view …
  - Complexity is the main cause of software faults
  - Model refinements introduce more complexity into the simulation codes (Example: from 2D to 3D)
  - Codes contain an unknown number of software faults
- How to improving the reliability of dispersion codes?
- How to increasing the trustworthiness of dispersion simulation results?

Software Reliability Engineering (SRE)

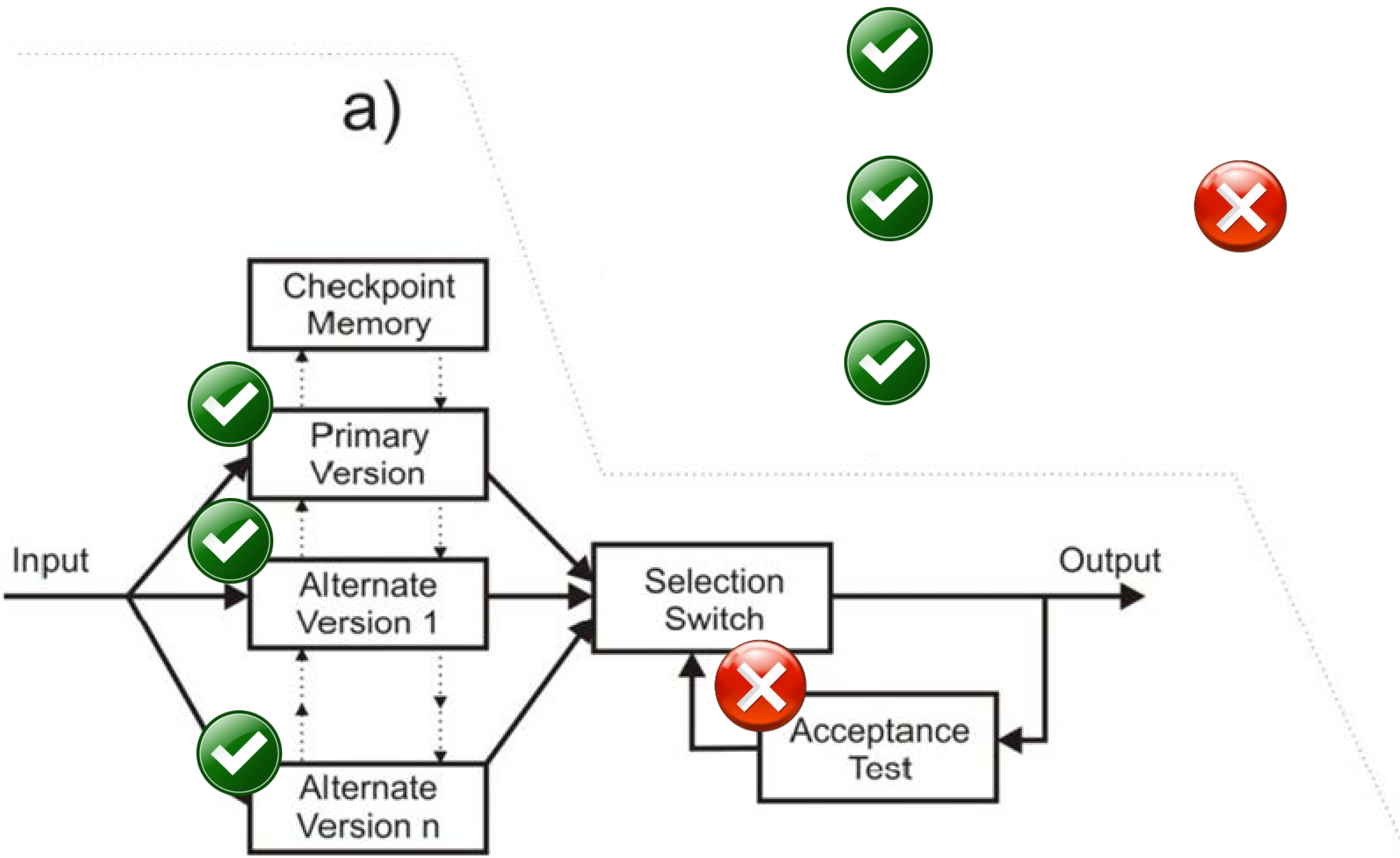
- Single version software reliability methods
  - Fault prevention
    - Coding standards, software metrics
  - Fault removal
    - Unit and integration testing
  - → **Initial verification of codes**



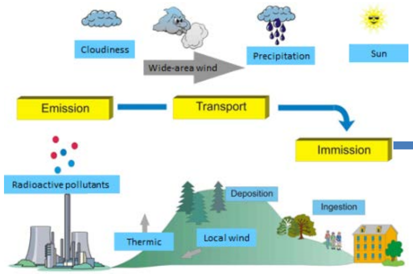
- Multi-version software reliability methods
  - Error detection and confinement
  - Acceptance tests
  - Recovery Blocks
  - → **Continuous verification of codes**

a.) **Recovery Blocks** and b.) **N-Version Programming**

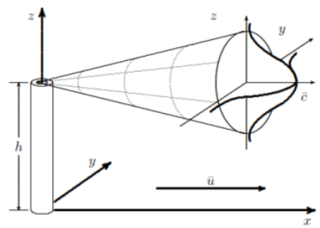
Rationale: software build differently is more likely to fail differently.



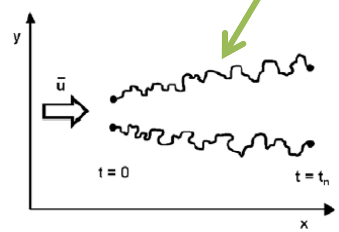
From Models to Simulation Codes



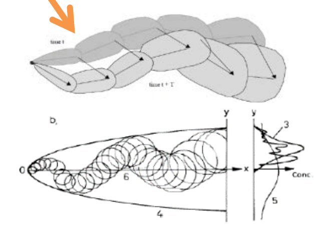
$$\frac{\partial \bar{c}}{\partial t} + \bar{u}_k \frac{\partial \bar{c}}{\partial x_k} = -\frac{\partial \overline{u'_k c'}}{\partial x_k} + \bar{P}_c$$



Gaussian Plume Model



Lagrange Particle Model



Puff Models

Simulation code C_1

Simulation code C_2

Simulation code C_3

Input data and parameters: *[P, d_P]*

Results C_1 *[R_1, d_R]*

Results C_2 *[R_2, d_R]*

Results C_3 *[R_3, d_R]*

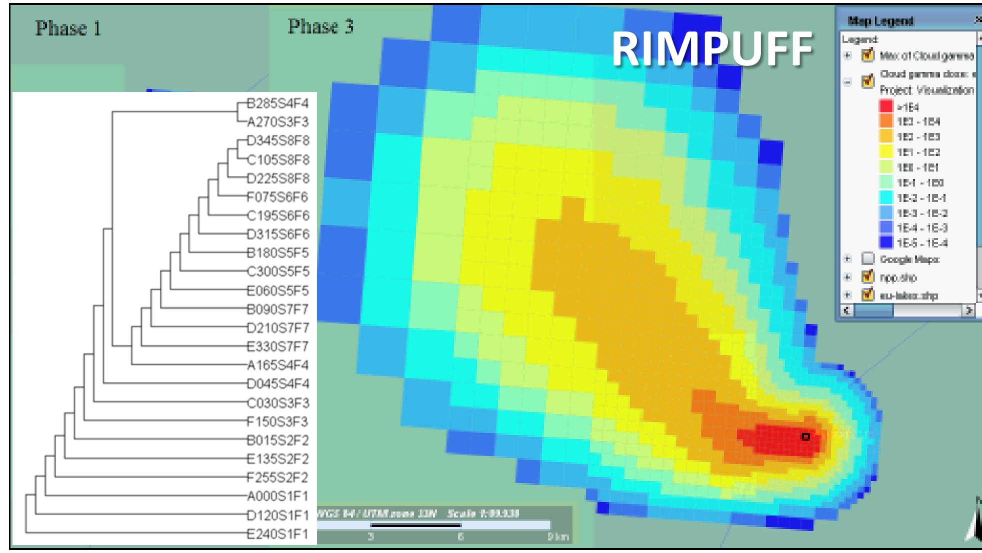
Disperison codes and N-Version Programming

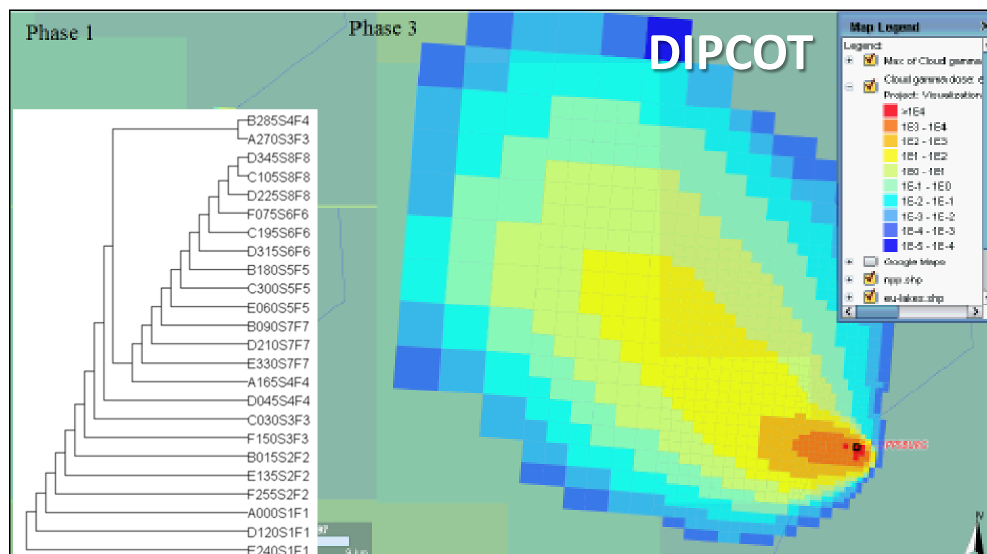
- **Example:** RODOS (Real-time Online Decision Support System for nuclear emergency management)
  - ATSTEP (Karlsruhe Institute of Technology)
    - Puff model with time-integrated elongated puffs
  - DIPCOT (National Centre for Scientific Research “Demokritos”, Greece)
    - Lagrange particle model
  - RIMPUFF (Risø DTU National Laboratory, Denmark)
    - Puff model

- Prerequisites for N-Version programming:
  - ✔ Developed by completely independent teams
  - ✔ Attempt to solve the transport equation numerically (same software requirements)

Comparing Dispersion Simulation Results

- Problem:
  - **How to compare individual results?**
  - Inherent model differences
  - Continuous mathematical space
- Idea:
  - **Compare taxonomies of results instead of individual results**
  - Discrete mathematical space





Reference Input Case Ensemble

- 24 input cases with varying parameters
  - Wind speed (1—8 m/s) and direction (0—360°)
  - Diffusion (A-F) and release category (F1—F8)

A000S1F1 B180S5F5
D120S1F1 C300S5F5
E240S1F1 E060S5F5

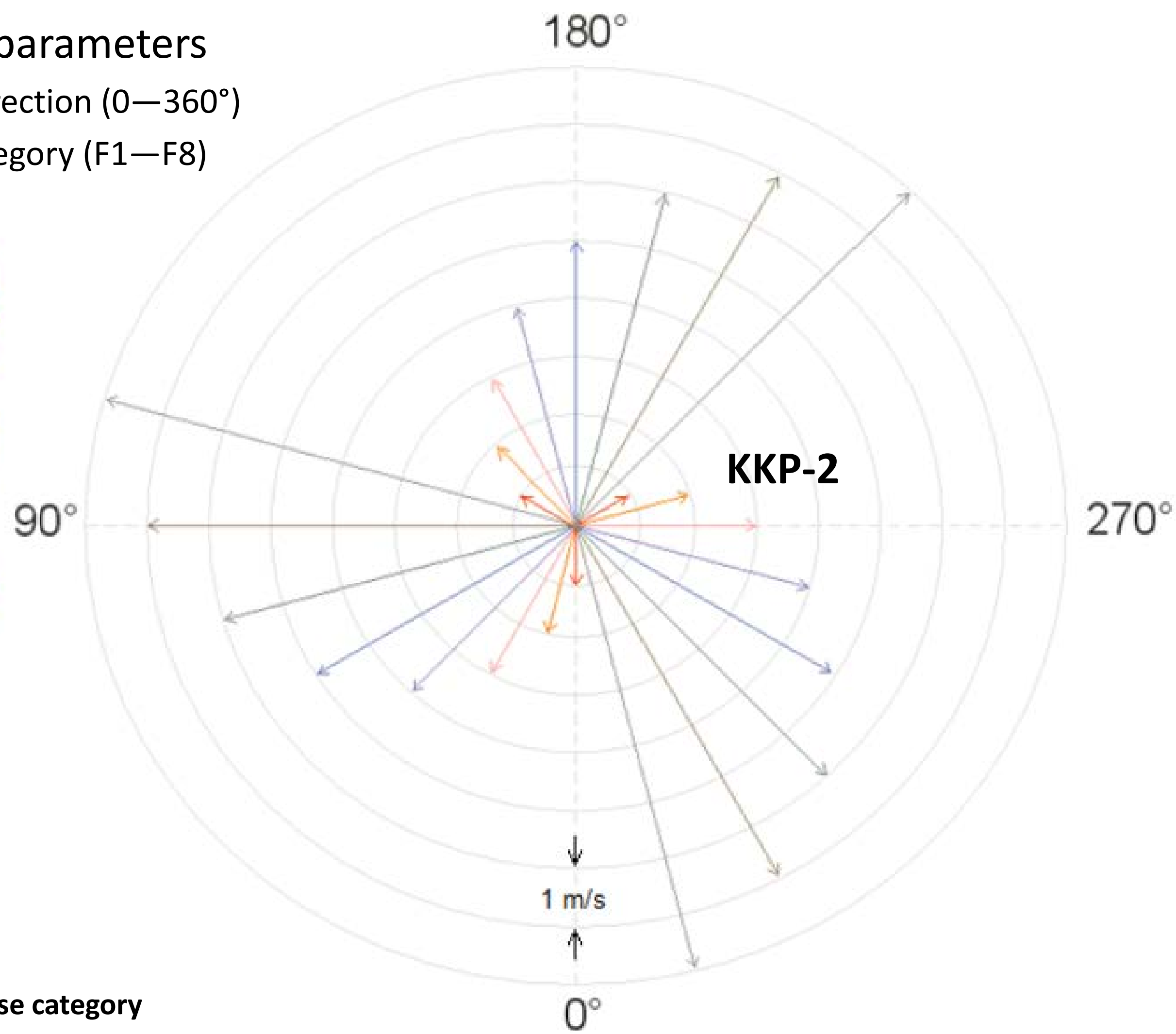
B015S2F2 C195S6F6
E135S2F2 D315S6F6
F255S2F2 F075S6F6

A270S3F3 B090S7F7
C030S3F3 D210S7F7
F150S3F3 E330S7F7

A165S4F4 C105S8F8
B285S4F4 D225S8F8
D045S4F4 D345S8F8

A 000 S1 F1

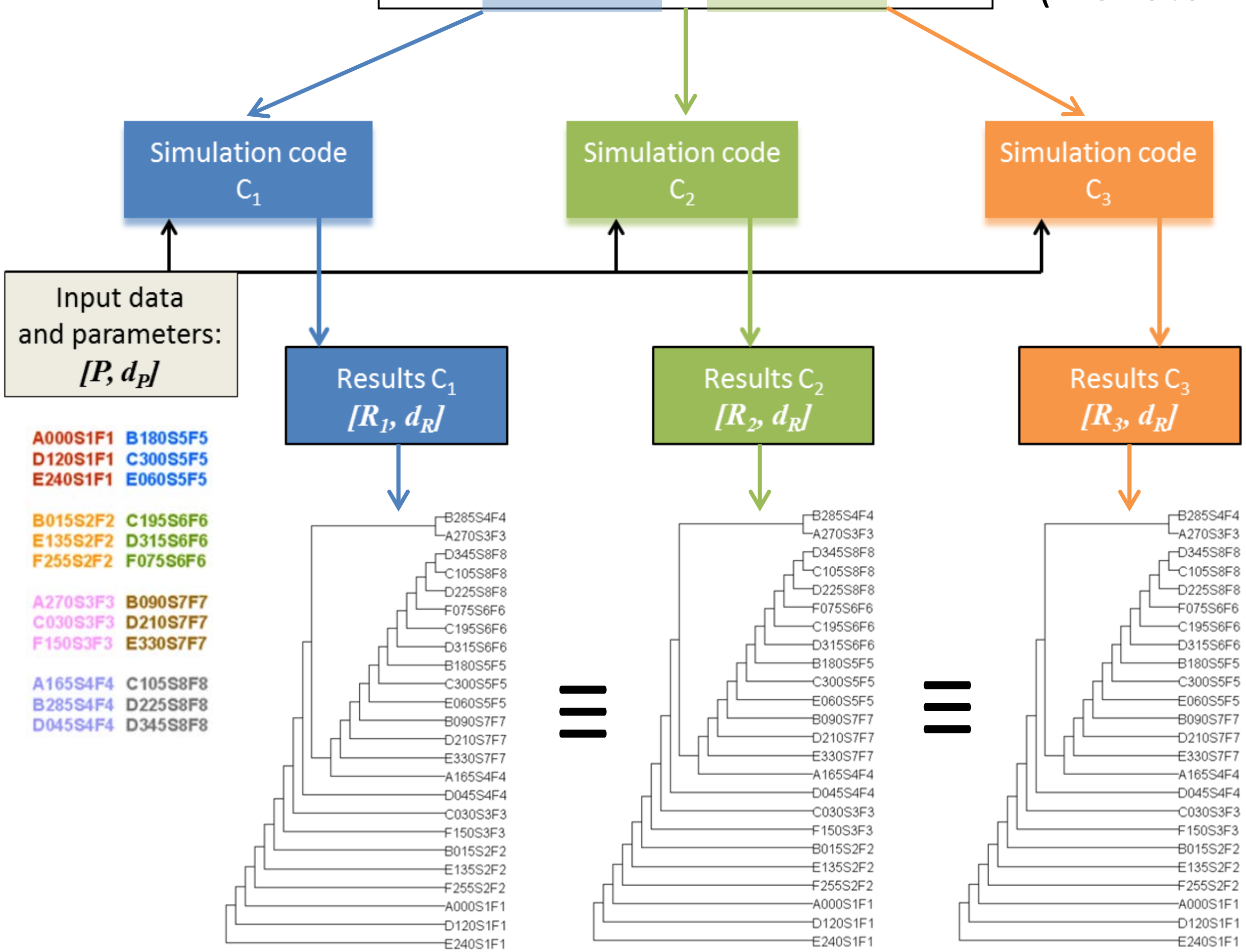
Diffusion category
Wind direction
Wind speed
Release category



A Voter for Dispersion Simulation Results

$$\frac{\partial \bar{c}}{\partial t} + \bar{u}_k \frac{\partial \bar{c}}{\partial x_k} = -\frac{\partial \overline{u'_k c'}}{\partial x_k} + \bar{P}_c$$

The transport equation is a **deterministic model** (monotonicity, distance preservation)



$C_k: [P, d_p] \rightarrow [R_k, d_R]$

Isometric map

(distance preserving function)

Taxonomies of results will be identical

if and only if

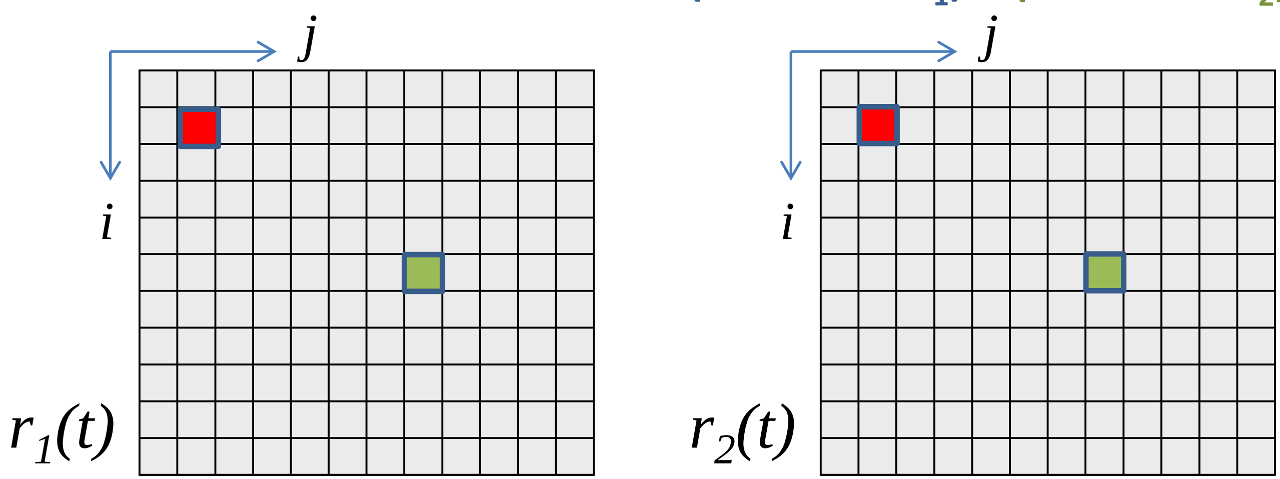
codes correctly implement models

A Metric for Dispersion Simulation Results

- **Dispersion simulation results** = distributions of continuous variables in a finite space

Residual sum of squares: $RSS = \sum_{i=1}^{n}[y_i - f(x_i)]^2$

Observed values (Results of C_1) | Estimated values (Results of C_2)



Generalized RSS: $d_R(r_1, r_2) = GRSS(r_1, r_2) = \sqrt{\sum_{t=1}^{t \le TS} \sum_{i=1}^{i \le N} \sum_{j=1}^{j \le M} (r_1[t,i,j]^p - r_2[t,i,j]^p)^2}$ $\quad 0 < p \le 1$

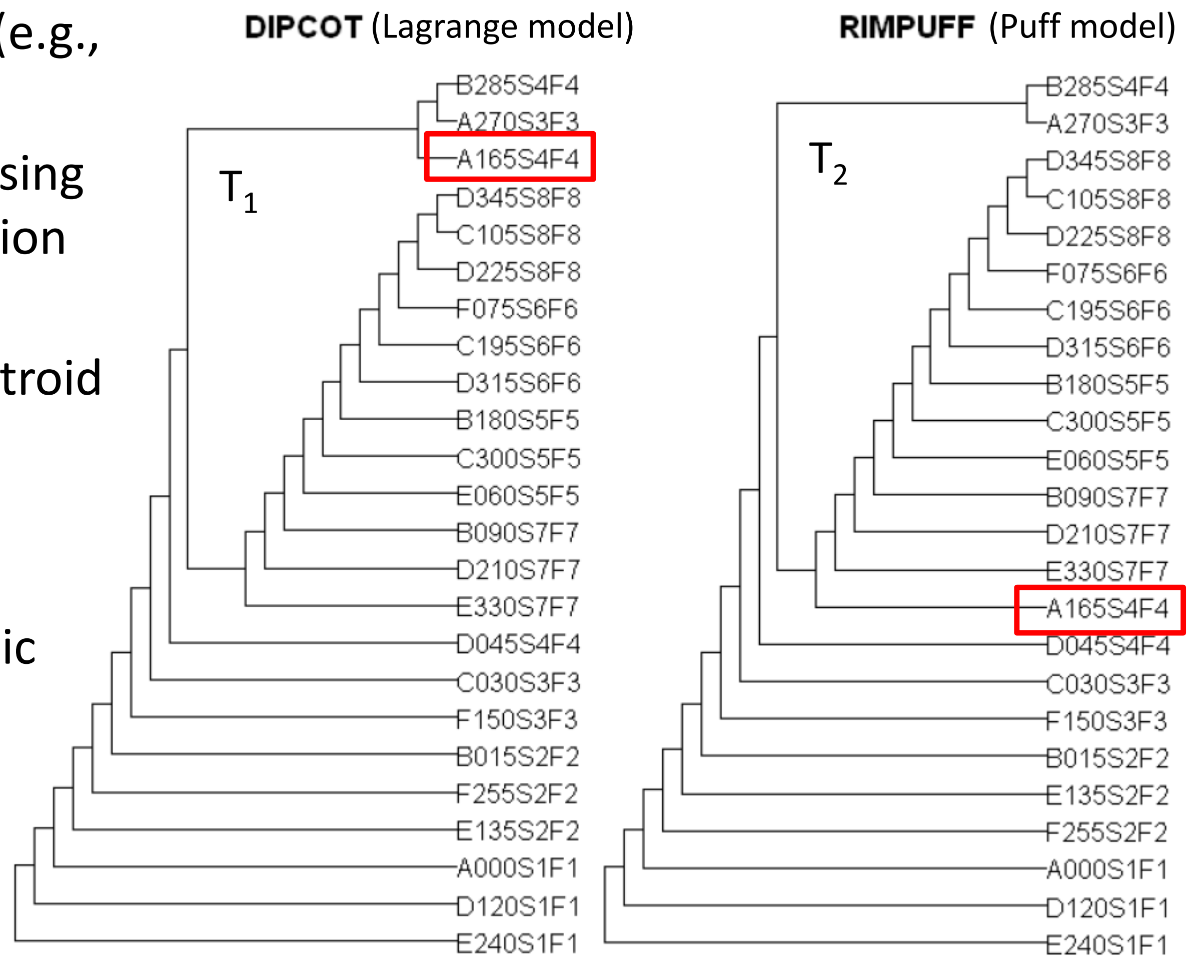
$p < 0.5$ → concentration of emitted substance is dominant

$p \ge 0.5$ → wind speed and direction are dominant

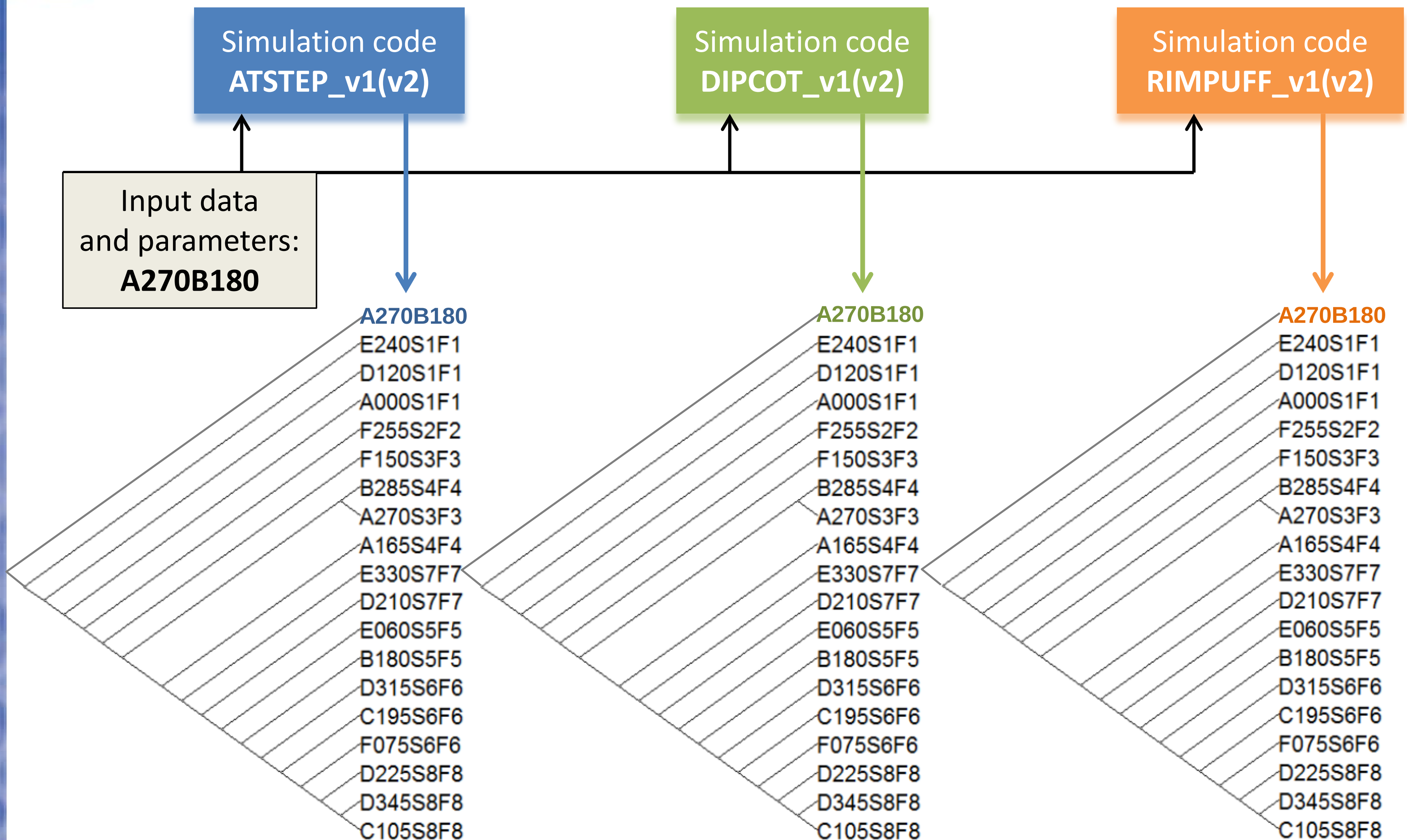
Building and Comparing Taxonomies of Results

1. Run simulations on reference input case ensemble using different simulation codes (e.g., ATSTEP, DIPCOT, RIMPUFF)
2. Compute distance matrix using GRSS on dispersion simulation results
3. Hierarchical clustering (centroid method) → taxonomies of results
4. Compute the topological distance between taxonomic trees: $RF(T_1,T_2)$
   - max RF(N) = 2N-6
   - max RF(24) = 42

$$RF\ (DIPCOT,\ RIMPUFF,\ GRSS_{p=0.6}) = 4\ (9.52\%)$$

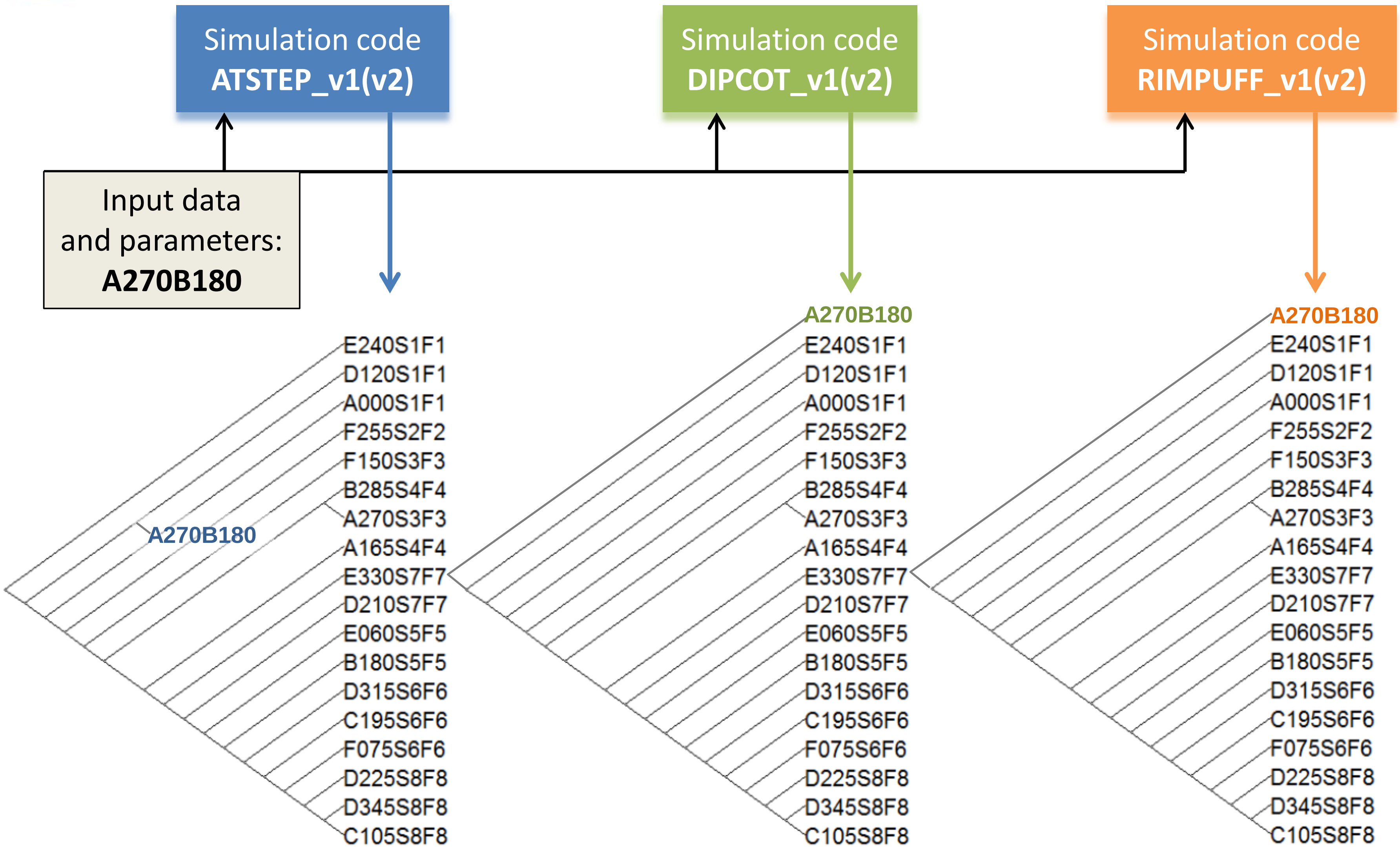


A Taxonomy-Based Voter for Dispersion Simulation Results



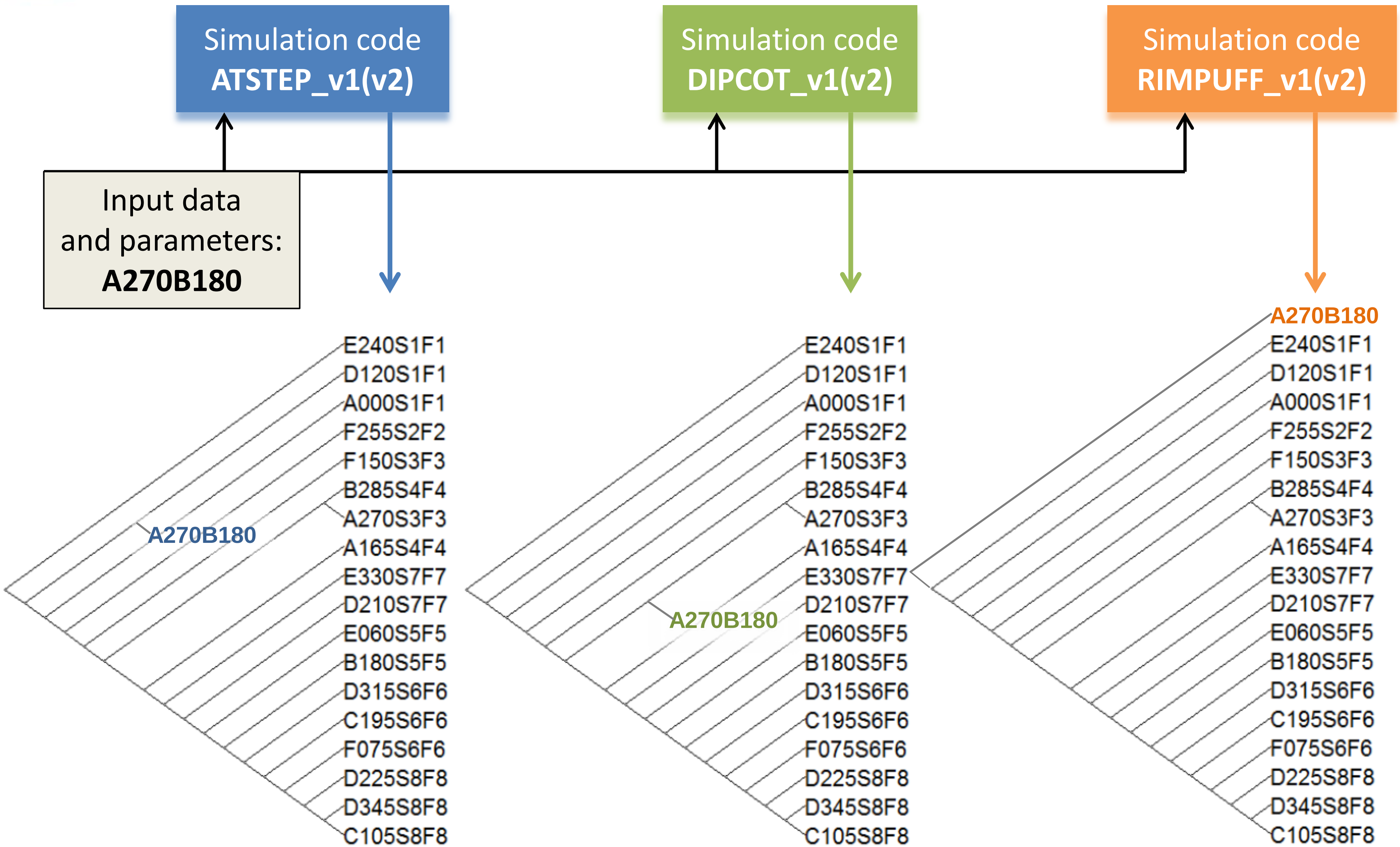
Strict Consensus

A Taxonomy-Based Voter for Dispersion Simulation Results



Majority Consensus

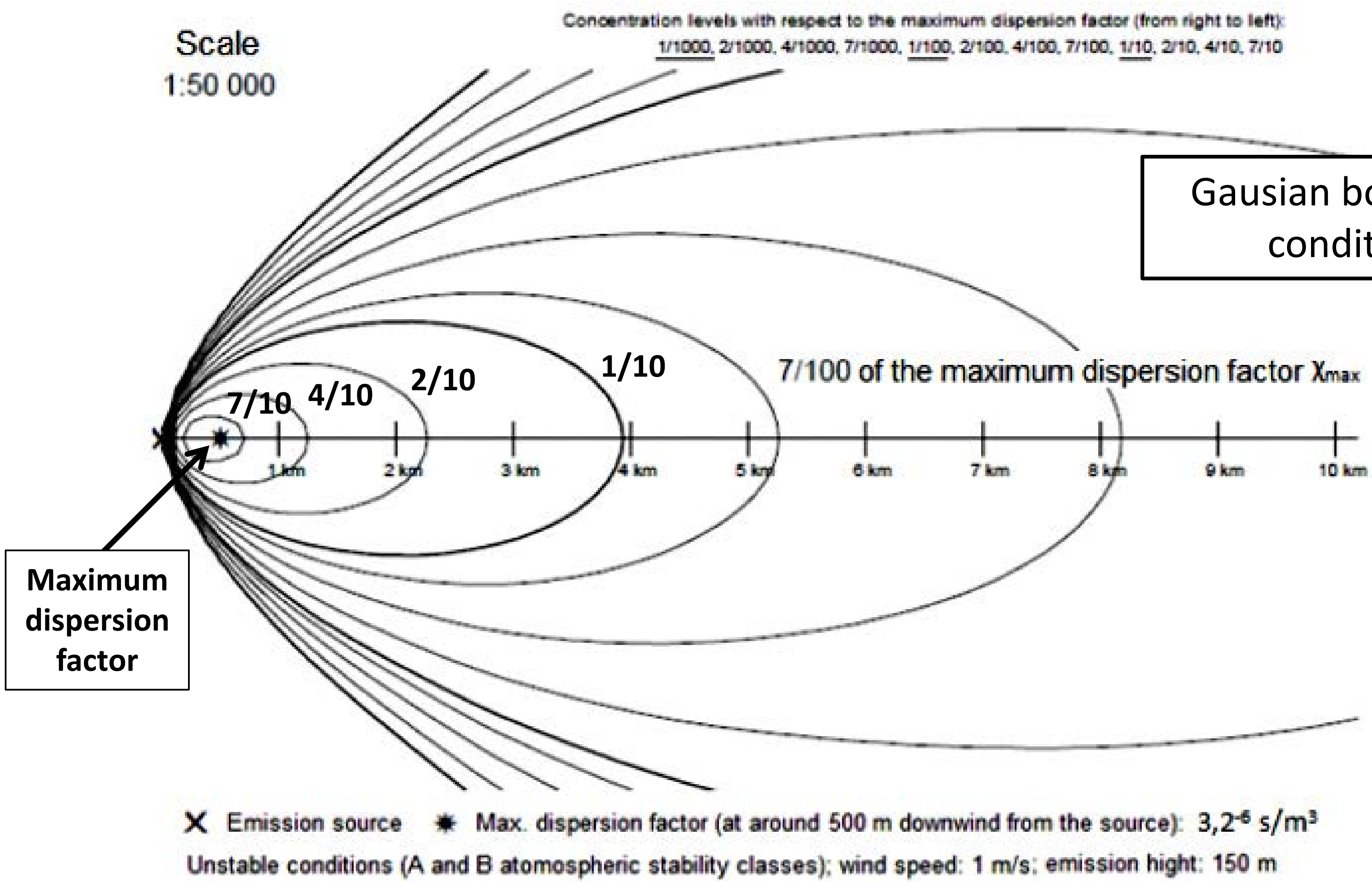
A Taxonomy-Based Voter for Dispersion Simulation Results



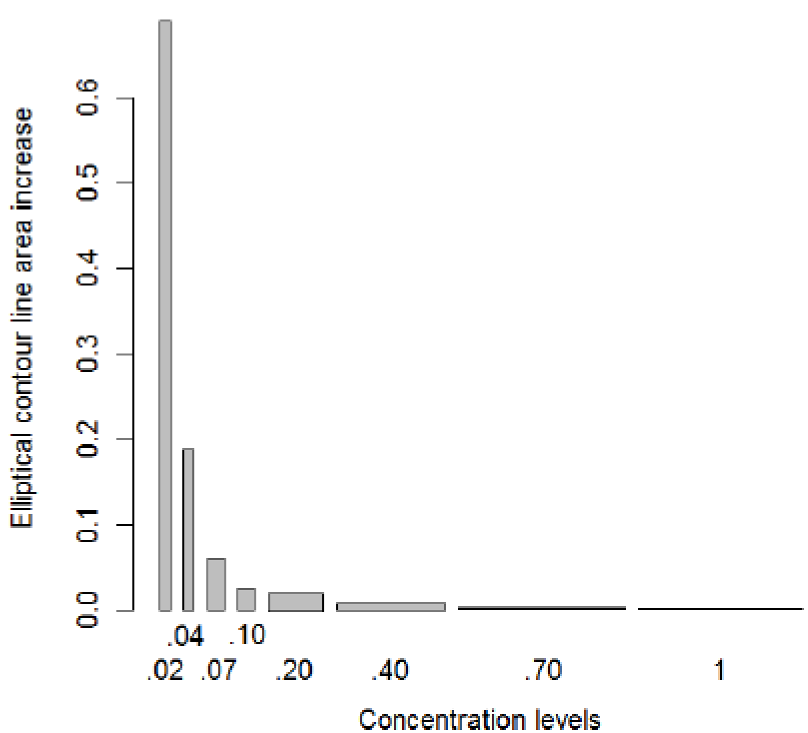
Disagreement

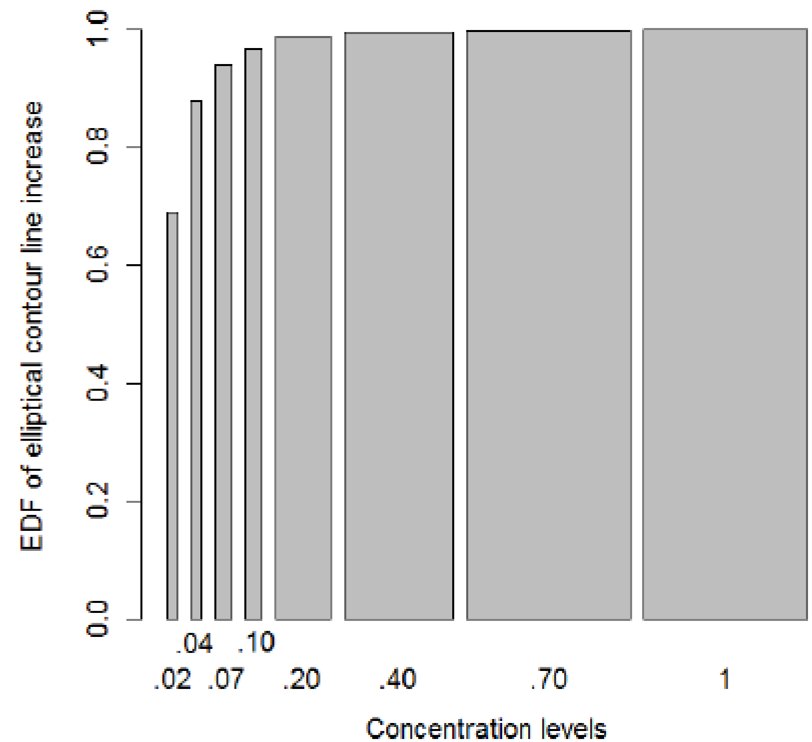
An Acceptance Test for Dispersion Simulation Results

Source: *„Leitfaden für den Fachberater Strahlenschutz der Katastrophenschutzleitung bei kerntechnischen Notfällen“*



An Acceptance Test for Dispersion Simulation Results





Weibull distribution: $F_0(x) = 1 - e^{-(x/\lambda)^k}$

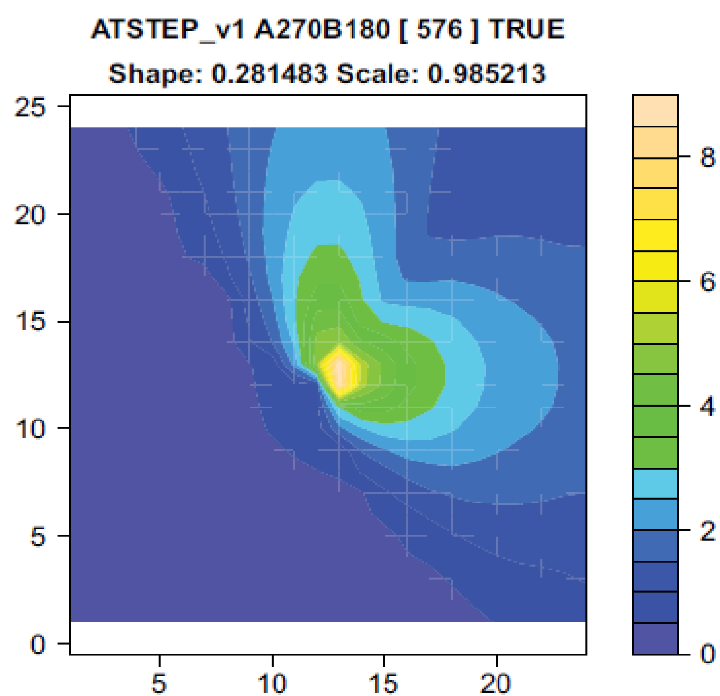
k Shape parameter of the Weibull distribution

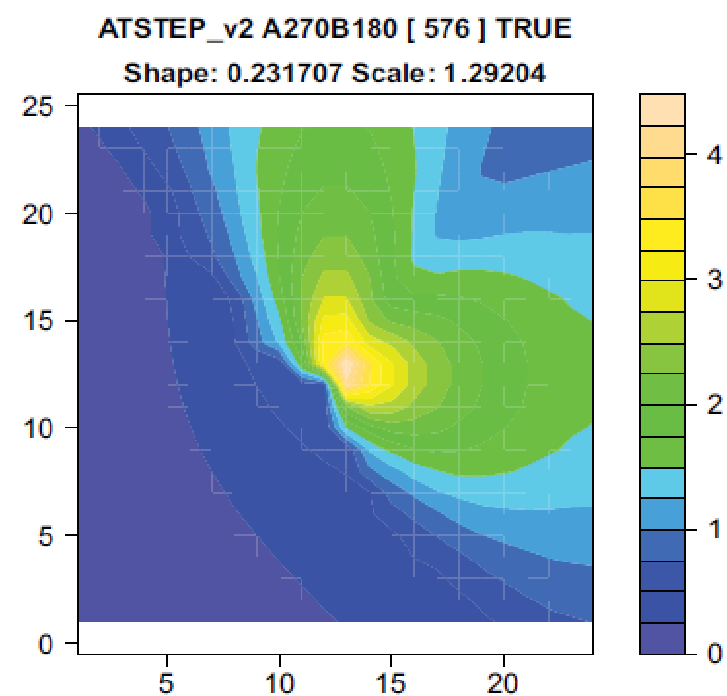
λ Scale parameter of the Weibull distribution

Acceptance test: Check if empirical distribution (from results) fits the hypothetical distribution by means of a goodness of fit test (Kolmogorov-Smirnov).

Experimental Validation of the Approach

- Two versions of the RODOS system
  - RODOS_v1 (ATSTEP_v1, DIPCOT_v1, RIMPUFF_v1)
    - Bug fixing and improvements brought to the system
  - RODOS_v2 (ATSTEP_v2, DIPCOT_v2, RIMPUFF_v2)
- Reference input case ensemble (24 cases)
- Extended input case ensemble – with rotating wind (24 cases)





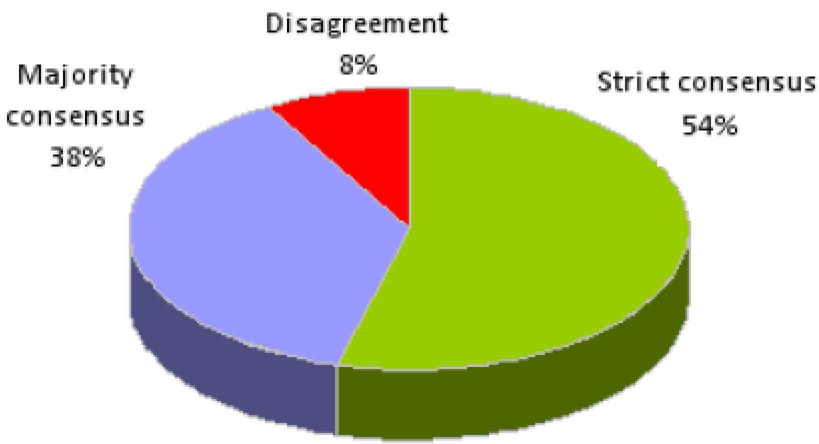
Results: Kolmogorov-Smirnov Test

Table 1. Kolmogorov-Smirnov test results for the two input case ensembles for RODOS_v1 and RODOS_v2.

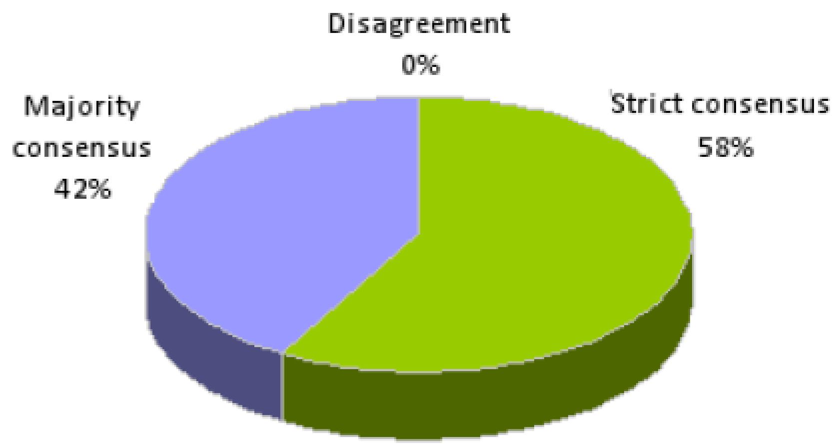
			Reference Input Case Ensemble			
$\alpha = 1\%$	**ATSTEP_v1**	**ATSTEP_v2**	**DIPCOT_v1**	**DIPCOT_v2**	**RIMPUFF_v1**	**RIMPUFF_v2**
Passed	16 / 24	22 / 24	20 / 24	19 / 24	21 / 24	21 / 24
Percent	66.67%	**91.67%**	83.33%	79.17%	87.50%	87.50%
			Second Input Case Ensemble			
Passed	17 / 24	20 / 24	15 / 24	19 / 24	20 / 24	23 / 24
Percent	70.83%	**83.33%**	62.50%	**79.17%**	83.33%	**95.83%**

- RODOS_v2
  - Improves over RODOS_v1 in 4 out of 6 cases
  - RIMPUFF handles wind rotation better than constant wind direction

Results: Taxonomy-Based Voter



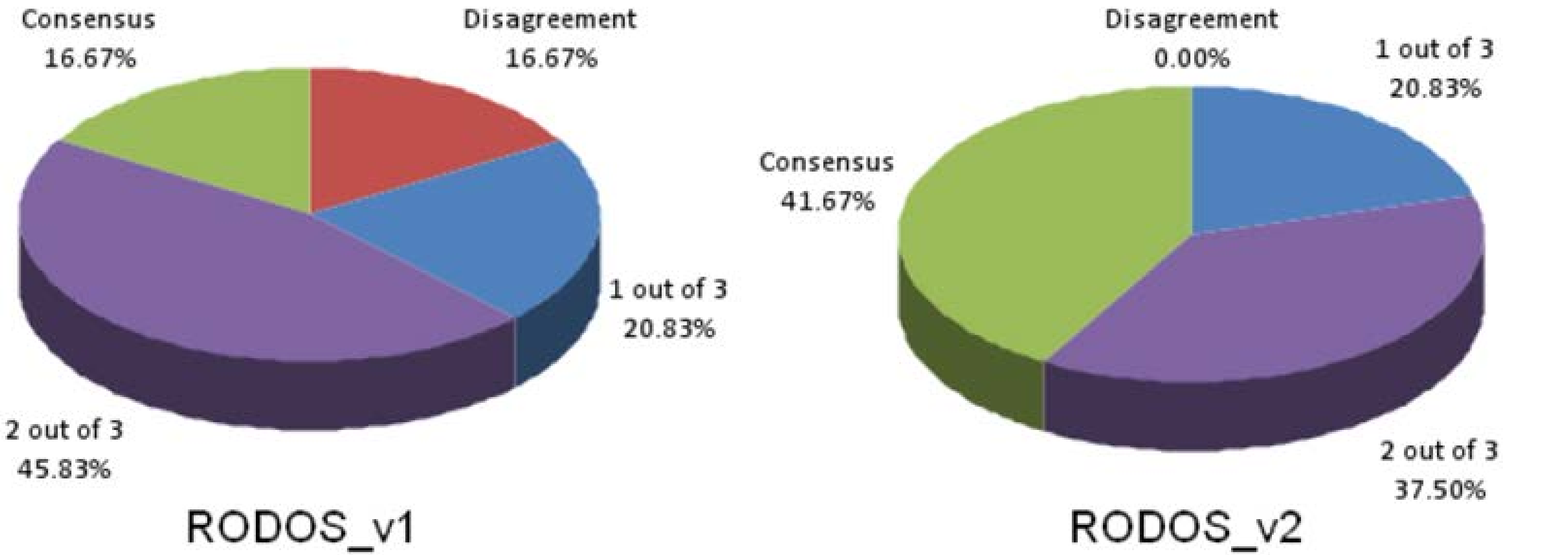
RODOS_v1



RODOS_v2

- RODOS_v2
  - No cases of disagreement
  - More cases of strict and majority consensus

Results: Taxonomy Voter & Acceptance Test



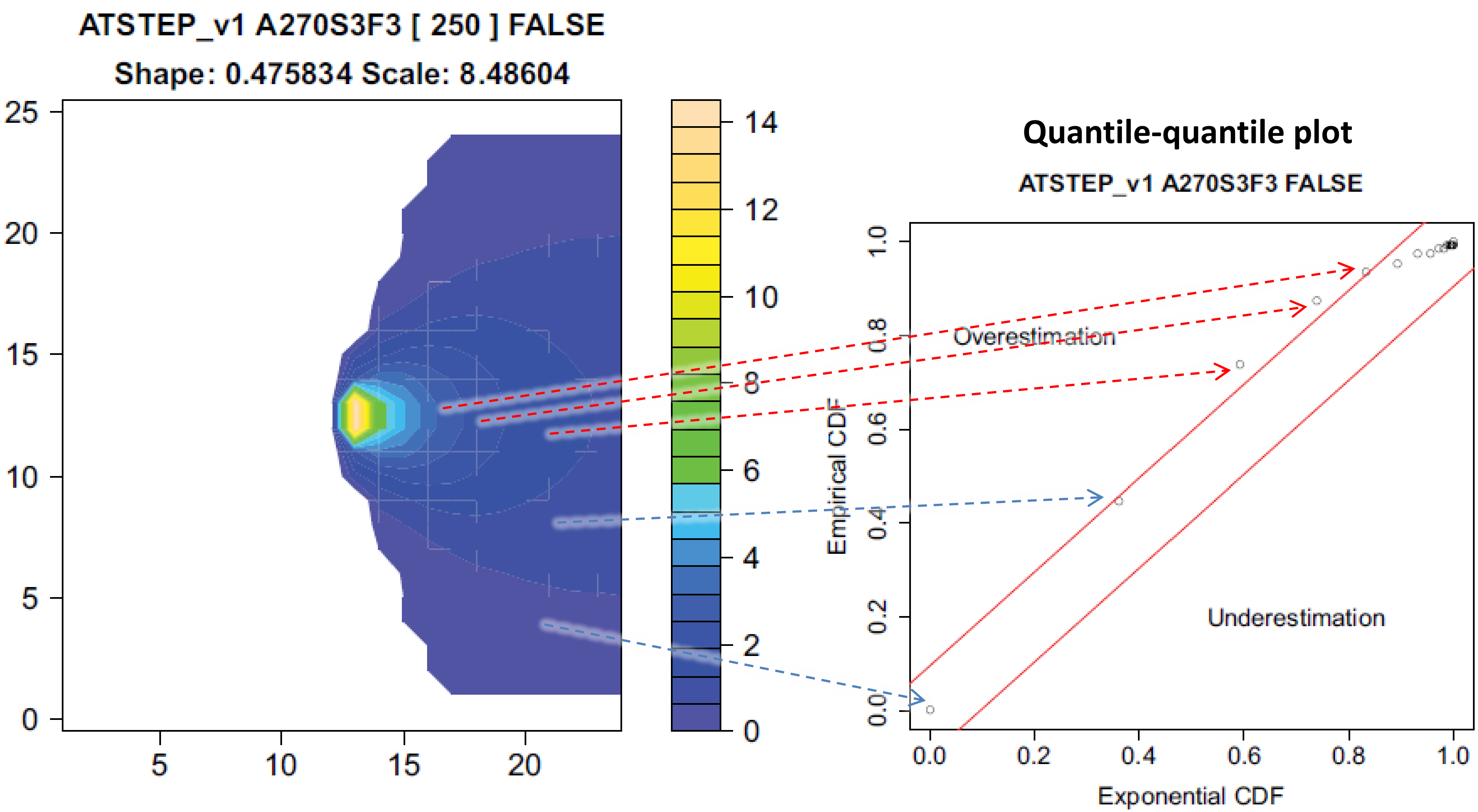
- RODOS_v2
  - No cases of disagreement
  - Many more cases of consensus

Conclusion

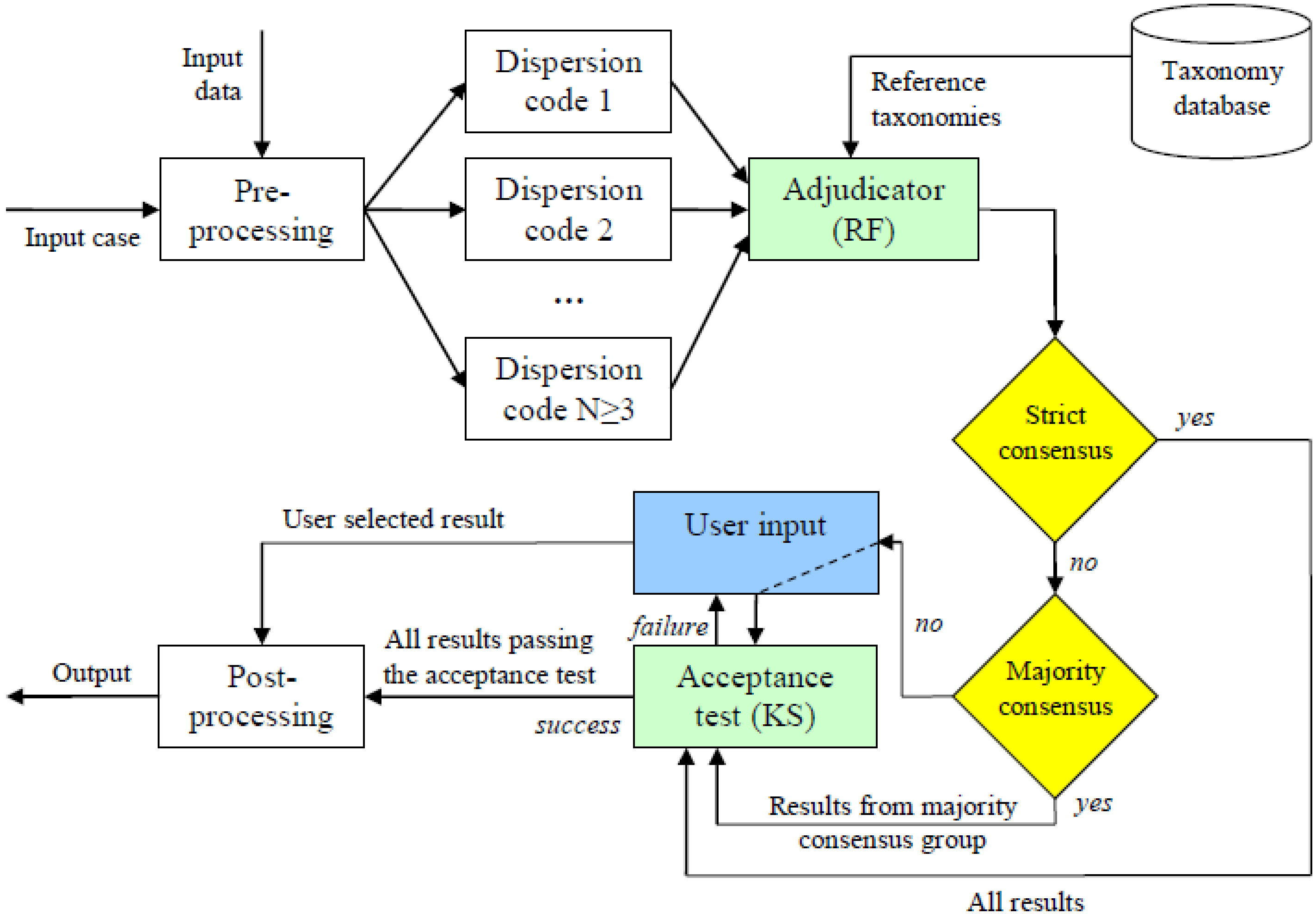
- The **acceptance test** and the **taxonomy-based voter** are sensitive to the improvements brought to the RODOS system
- Fully automated software metrics for the continuous verification of dispersion simulation codes
- Complementary to visual inspection and other means of verification
- Can be used for
  - Model/code inter-comparisons based on many input cases
  - Ensemble-based dispersion modelling systems
- Source code available

Visualization of Kolmogorov-Smirnov Test Results

- Generated in ***R*** (statistical computing)



- Workflow for a dispersion forecasting system supporting functionally redundant simulation codes and continuous verification of results.



Monotonicity of Time-Integrated Doses

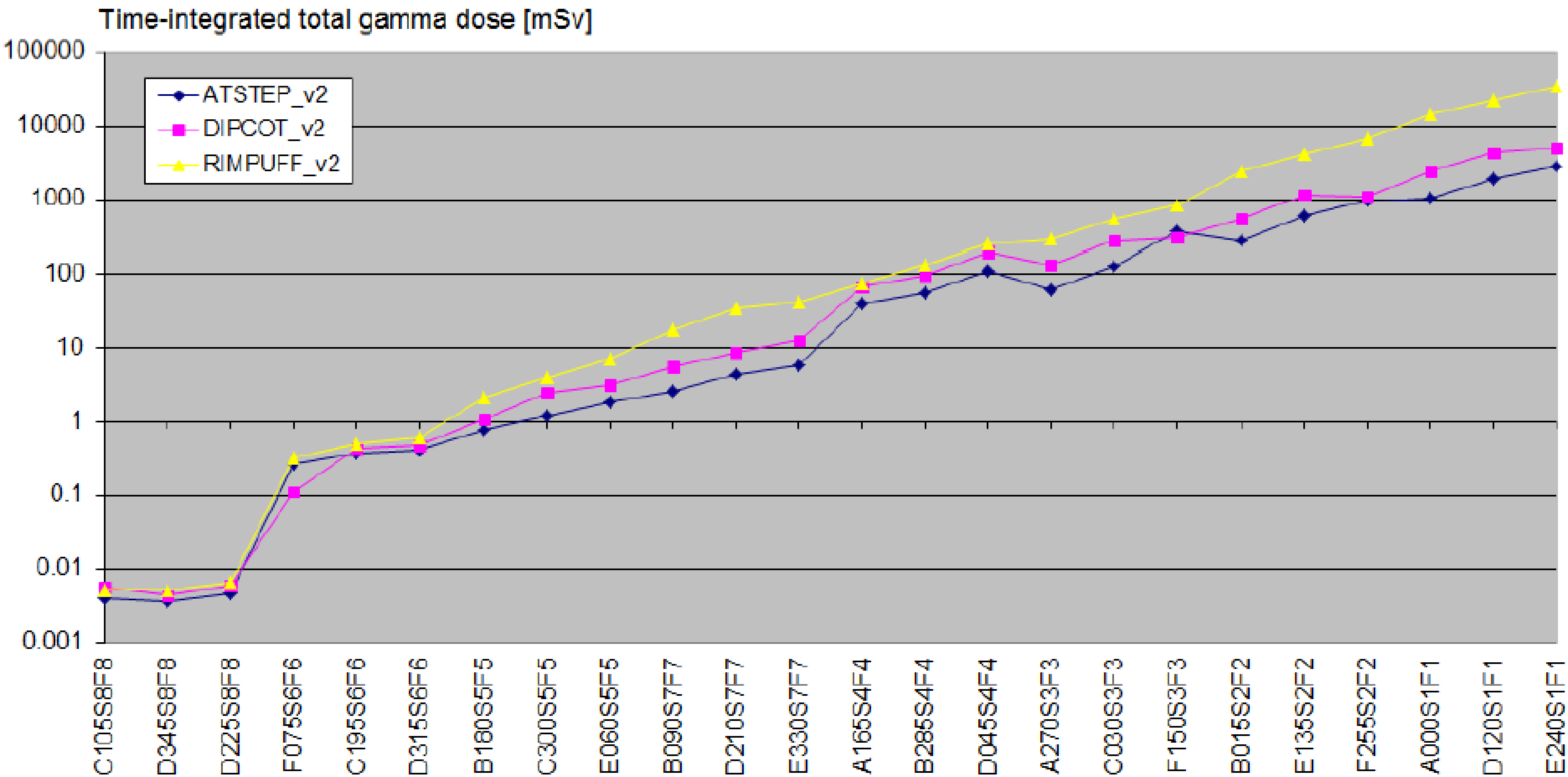


Figure 29: The time-integrated gamma dose for all input cases as computed by ATSTEP, DIPCOT, and RIMPUFF (corresponding to RODOS_v2). The values are summed up over the entire monitored area.