VariantInc: Automatically Pruning and Integrating Versioned Software Variants

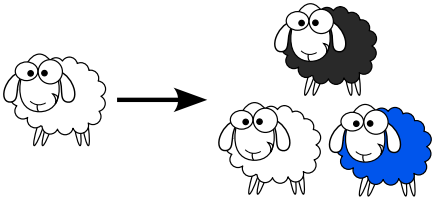
Sebastian Krieter, Jacob Krüger, Thomas Leich, Gunter Saake | August 30, 2023 — SPLC'23

Software Engineering
Programming Languages

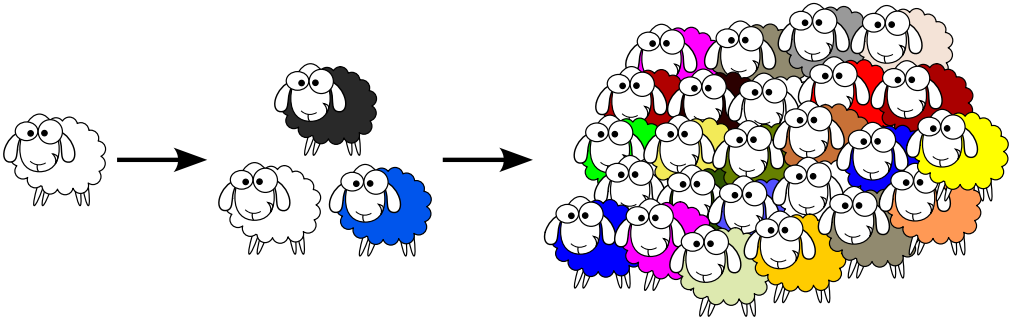
universität ulm

Fork-based Clone-And-Own

Fork-based Clone-And-Own



Fork-based Clone-And-Own



VariantInc – Envisioned Use Cases

- **What features are developed in forks?**
  1. Simplify and analyze revision histories of forks
  2. Identify variations in forks

VariantInc – Envisioned Use Cases

- **What features are developed in forks?**
  1. Simplify and analyze revision histories of forks
  2. Identify variations in forks

- **Can we reintegrate externally developed features?**
  3. Automatically integrate forks into a platform
  4. Configure variations in space and time

VariantInc – Envisioned Use Cases

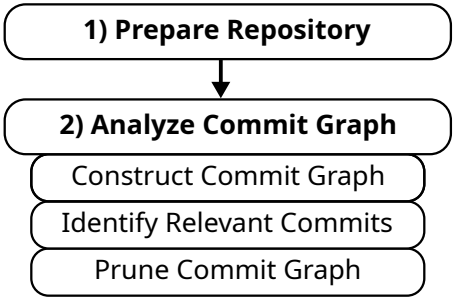
- **What features are developed in forks?**
  1. Simplify and analyze revision histories of forks
  2. Identify variations in forks
- **Can we reintegrate externally developed features?**
  3. Automatically integrate forks into a platform
  4. Configure variations in space and time
- **Can we migrate a fork-based SPL towards a variational control system?**
  5. Migrate towards variation control

VariantInc – Process

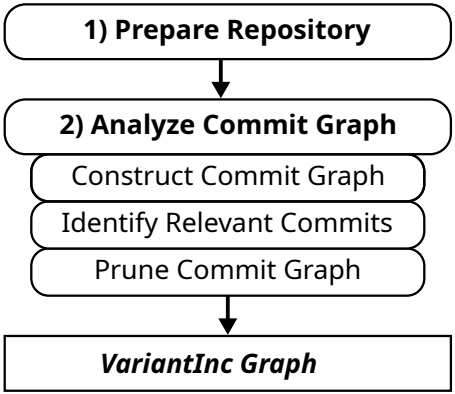
VariantInc – Process

1) Prepare Repository

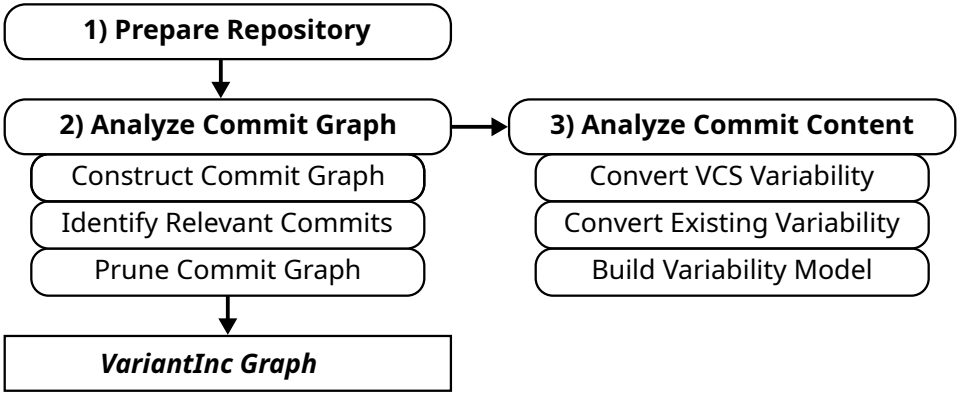
VariantInc – Process



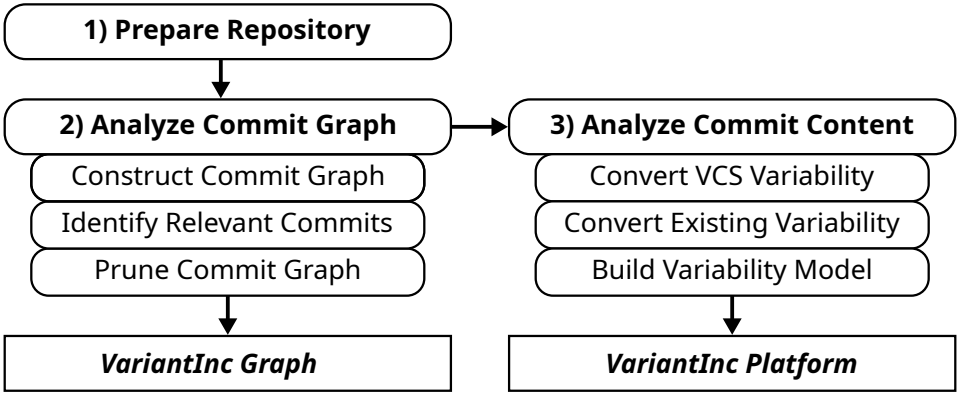
VariantInc – Process



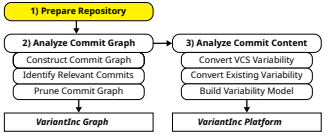
VariantInc – Process



VariantInc – Process

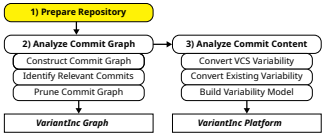


VariantInc Graph



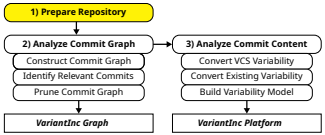
VariantInc Graph

- **Prepare repository**
  1. Search for publicly available forks on Github
  2. Clone main repository
  3. Fetch all commits from all available fork

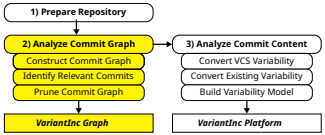


VariantInc Graph

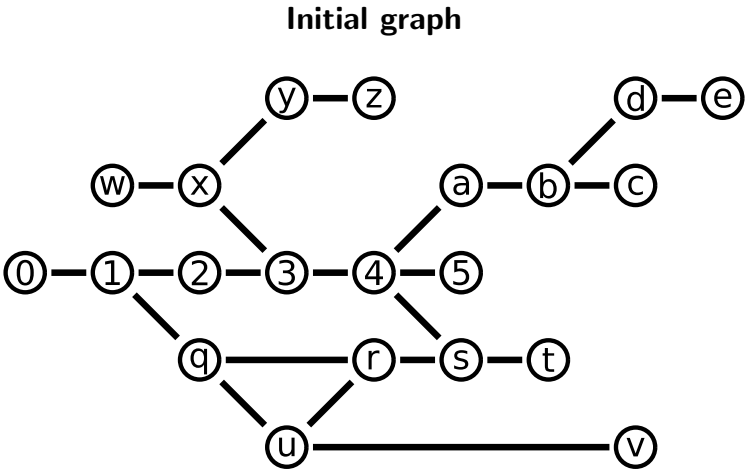
- **Prepare repository**
  1. Search for publicly available forks on Github
  2. Clone main repository
  3. Fetch all commits from all available fork



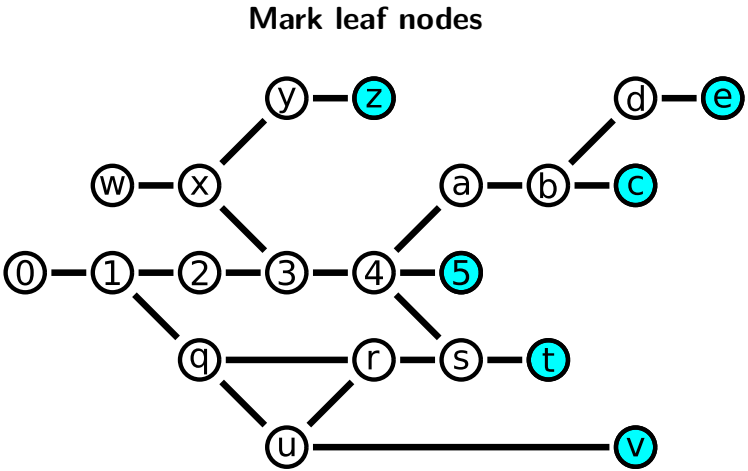
- **Analyze commit graph**
  1. Find a common root for all variants
  2. remove states that do not contribute variability



VariantInc Graph – Example

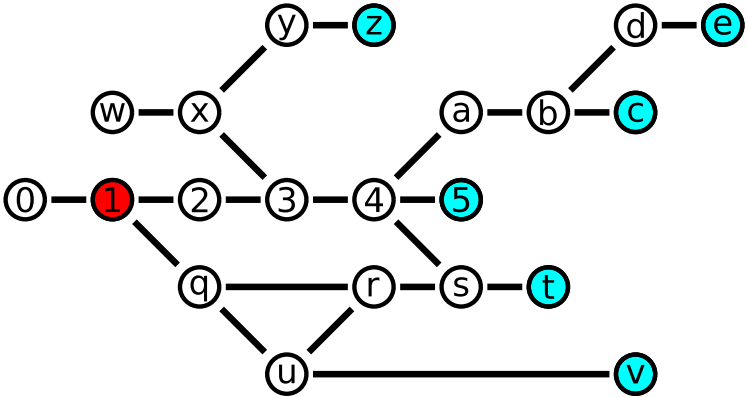


VariantInc Graph – Example

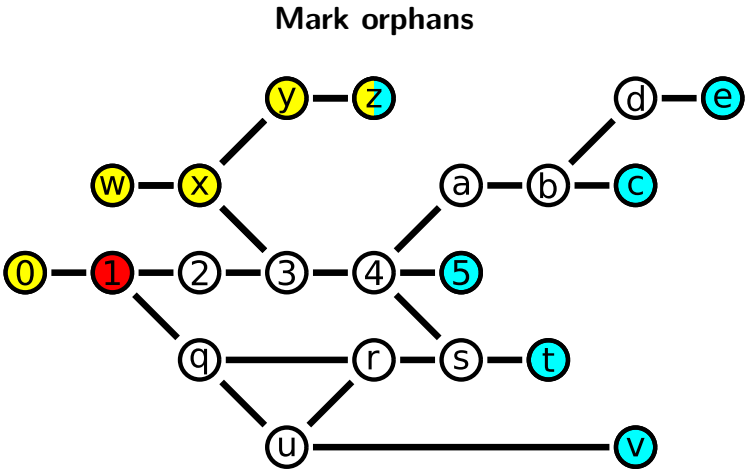


VariantInc Graph – Example

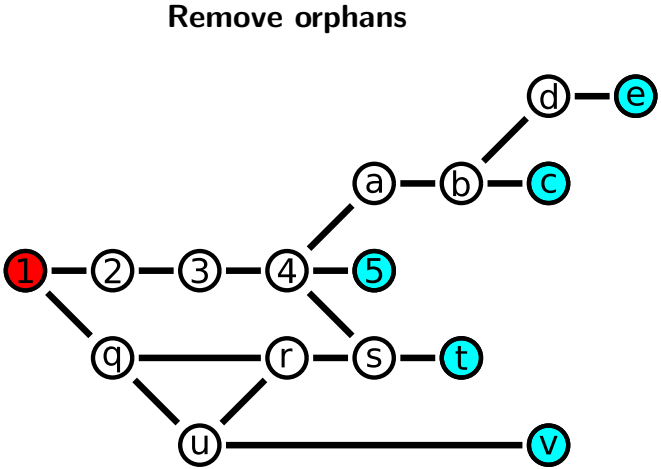
Compute most common ancestor



VariantInc Graph – Example

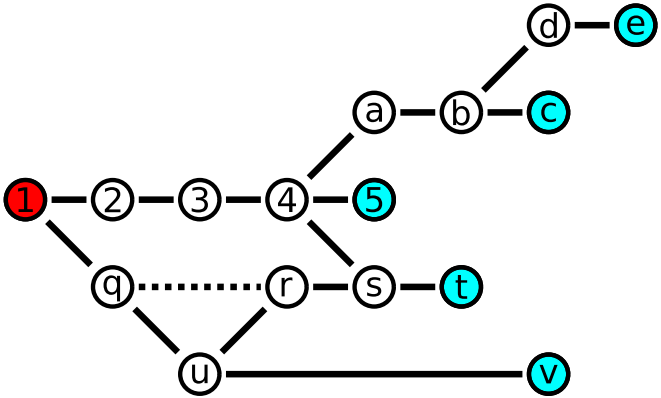


VariantInc Graph – Example



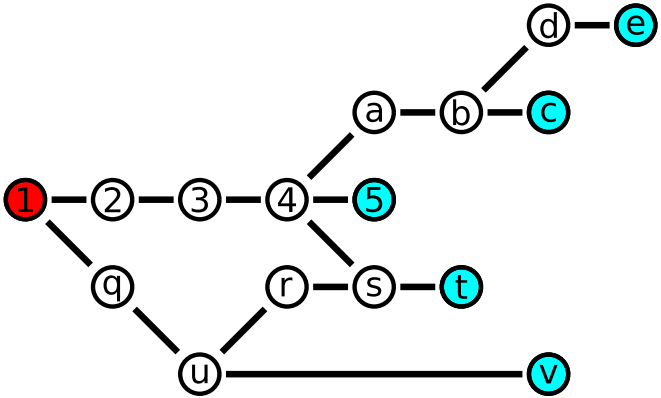
VariantInc Graph – Example

Mark transitive edges



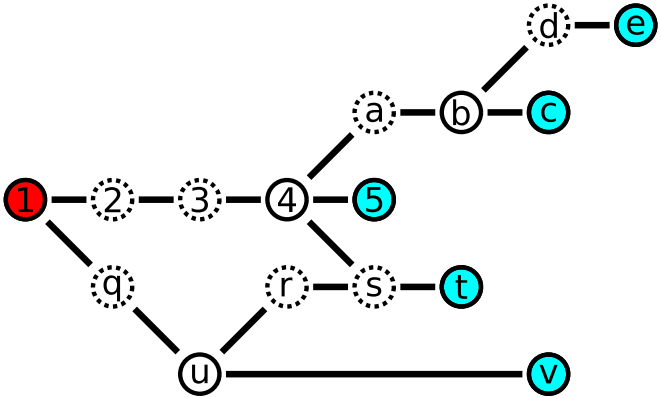
VariantInc Graph – Example

Remove transitive edges



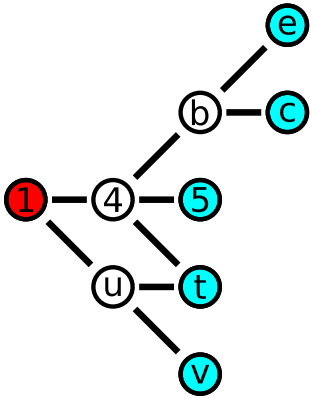
VariantInc Graph – Example

Mark commits with a single child

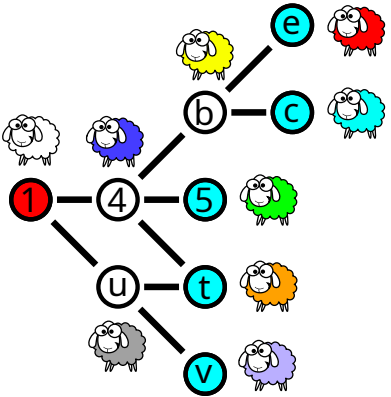


VariantInc Graph – Example

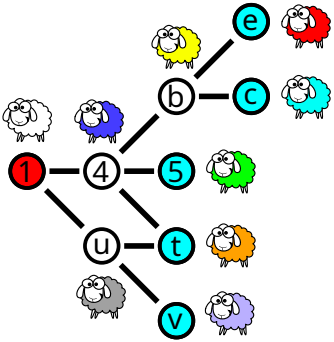
Remove commits



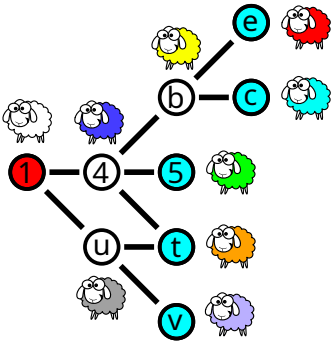
VariantInc Graph – Example



VariantInc Graph

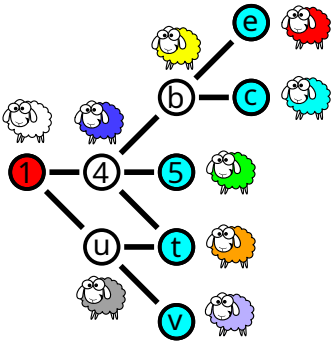


VariantInc Graph



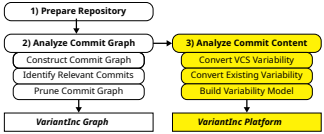
- Offering means to **analyze** and compare co-evolving **variants**

VariantInc Graph



- Offering means to **analyze** and compare co-evolving **variants**
- Provides support for developers to **decide which forks are relevant** for a platform

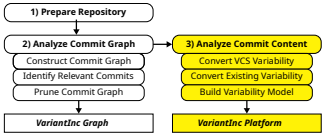
VariantInc Platform



VariantInc Platform

- **Analyze commit content**
  - Start with common ancestor and perform a breadth-first search
  - For each commit:
    - Compute **commit presence condition** for every line
    - Compute **variability presence condition** for every line
    - Combine both presence conditions

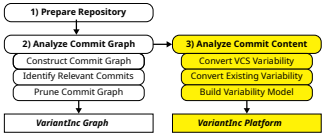
  $\Rightarrow$ Ordered list of lines with presence conditions

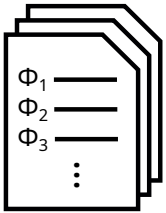


VariantInc Platform

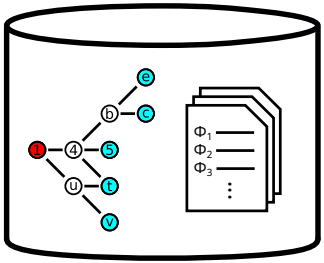
- **Analyze commit content**
  - Start with common ancestor and perform a breadth-first search
  - For each commit:
    - Compute **commit presence condition** for every line
    - Compute **variability presence condition** for every line
    - Combine both presence conditions

  $\Rightarrow$ Ordered list of lines with presence conditions

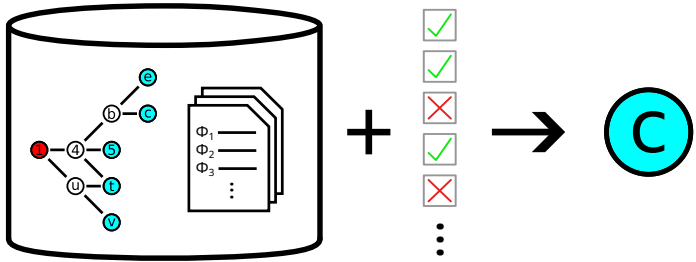




VariantInc Platform

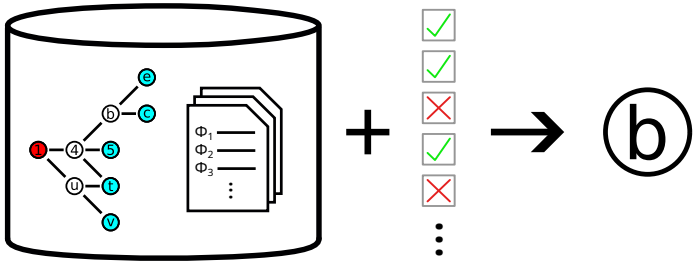


VariantInc Platform



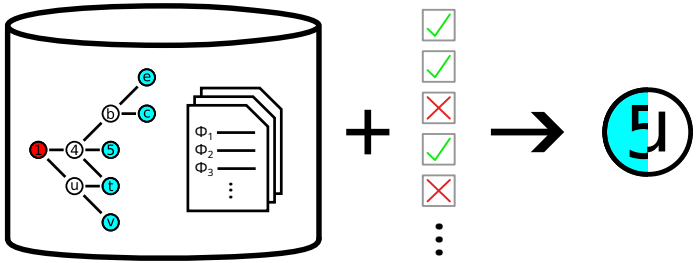
- **Configure variant from platform**
  - Evaluate presence conditions for each line

VariantInc Platform



- **Configure variant from platform**
  - Evaluate presence conditions for each line

VariantInc Platform



- **Configure variant from platform**
  - Evaluate presence conditions for each line

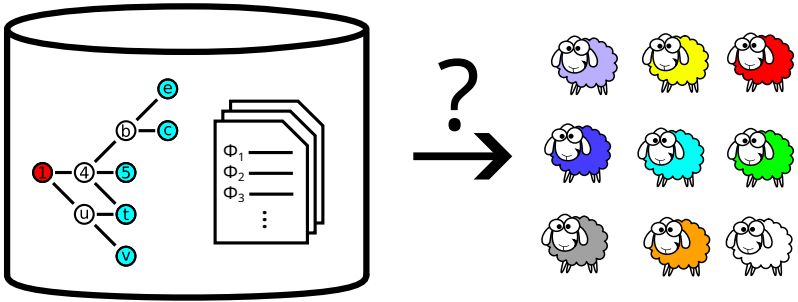
Evaluation Setup

- **Data**
  - 160 repositories from Github
    - #Forks: 2 – 3,944
    - #Branches: 1 – 3,592
    - #Commits: 27 – 150,562
    - #LOC: 2,228 – 21,562,320

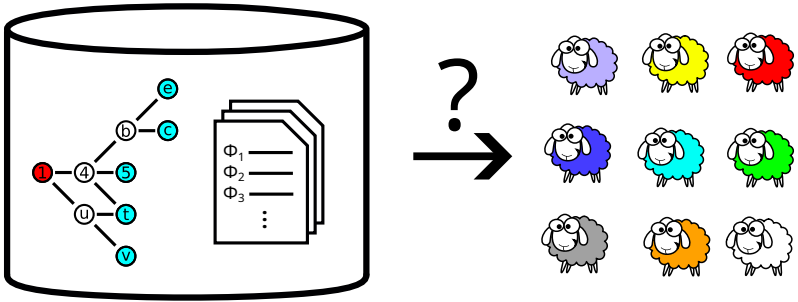
Evaluation Setup

- **Data**
  - 160 repositories from Github
    - #Forks: 2 – 3,944
    - #Branches: 1 – 3,592
    - #Commits: 27 – 150,562
    - #LOC: 2,228 – 21,562,320
- **Process**
  1. Build VariantInc Graph and Platform
  2. Analyze graph
  3. Test platform correctness

Insights – Correctness

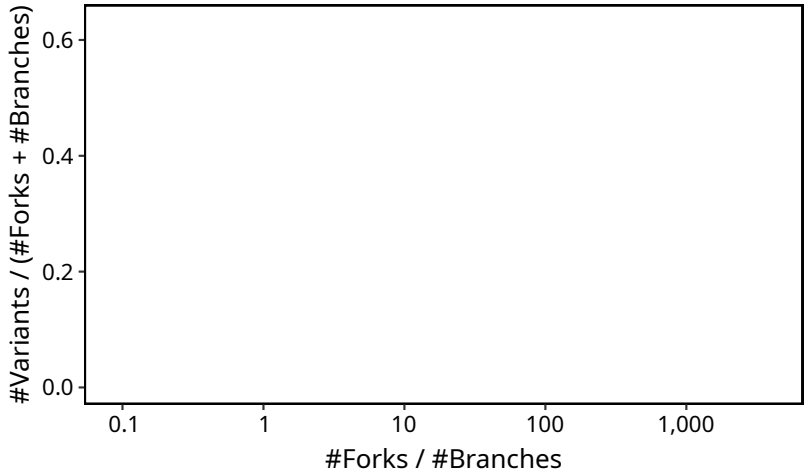


Insights – Correctness

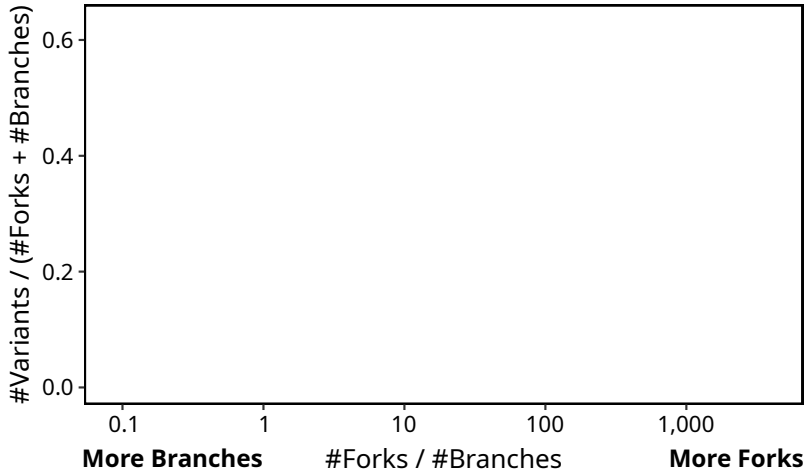


⇒ **All commits could be correctly configured using the platform**

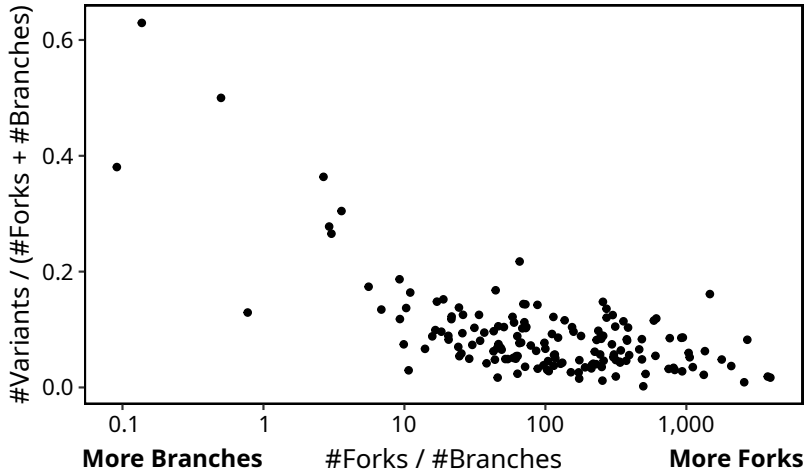
Insights – Branch-Based and Fork-Based Development



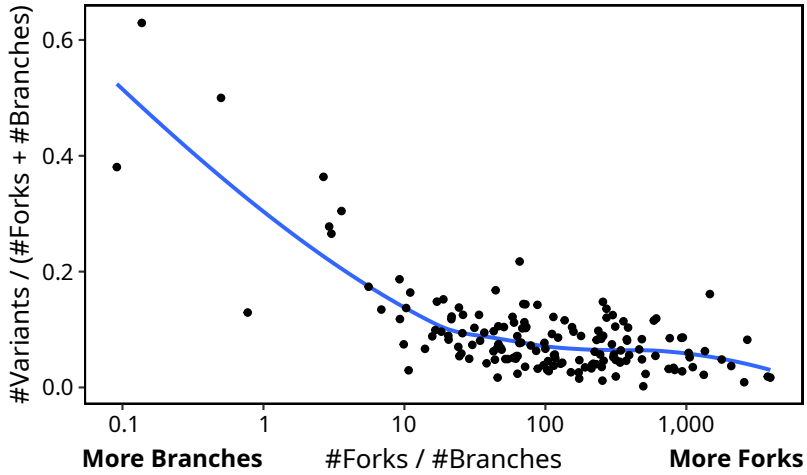
Insights – Branch-Based and Fork-Based Development



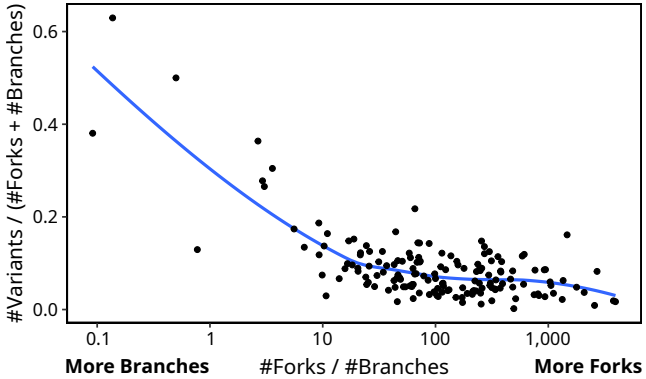
Insights – Branch-Based and Fork-Based Development



Insights – Branch-Based and Fork-Based Development

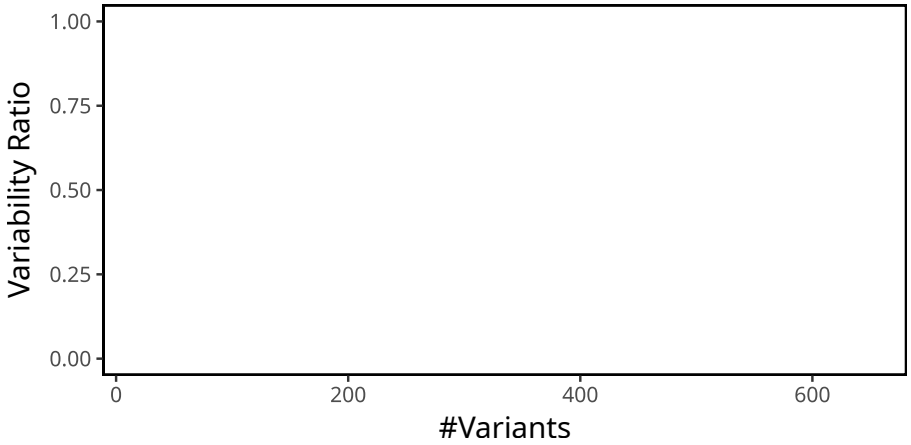


Insights – Branch-Based and Fork-Based Development

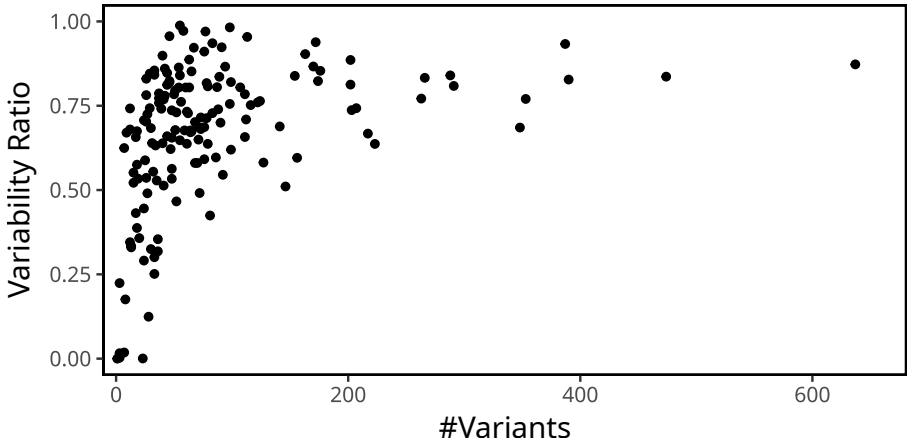


⇒ **Branches seem to cause more variants then than forks**

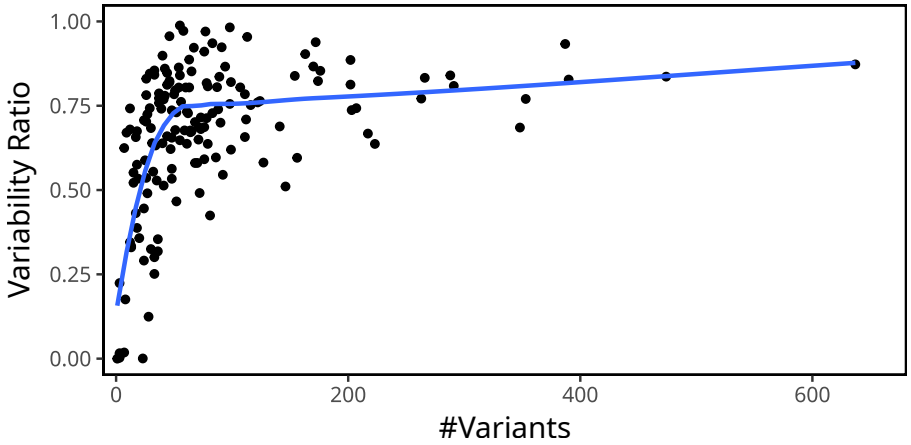
Insights – Variability Ratio



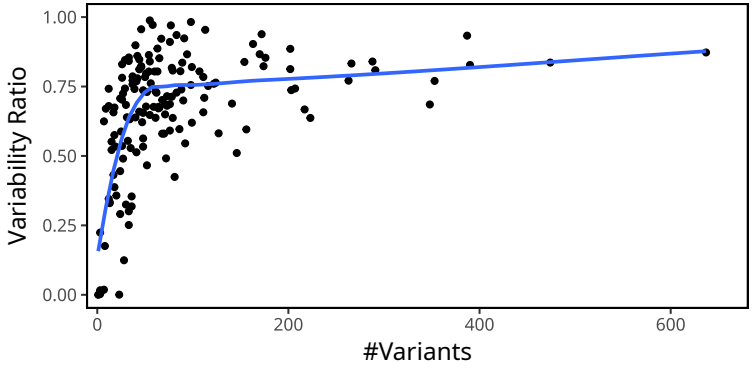
Insights – Variability Ratio



Insights – Variability Ratio



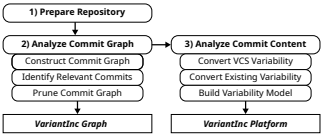
Insights – Variability Ratio

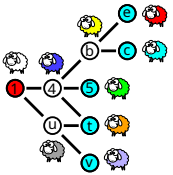


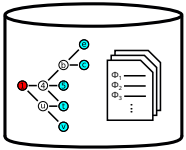
⇒ **Integrating more forks does not necessarily add more variability**

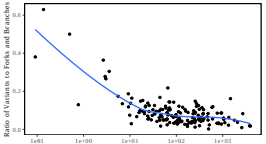
Conclusion

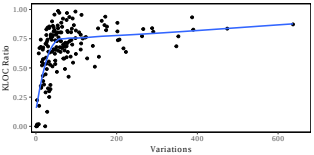
- **Introduction of VariantInc**
  - Enable novel analyses of revision histories
  - Facilitate the identification of unique variants in forks
  - Allows initial configuration of variants from revisions and existing variability











Conclusion

- **Introduction of VariantInc**
  - Enable novel analyses of revision histories
  - Facilitate the identification of unique variants in forks
  - Allows initial configuration of variants from revisions and existing variability
- Future combination with other tools for identifying features is desirable