

Scientific workflow management with ADAMS


Peter Reutemann¹ and Joaquin Vanschoren²

¹University of Waikato, Hamilton, NZ, fracpete@waikato.ac.nz

²Leiden University, Leiden, NL, joaquin@liacs.nl

 **Machine Learning**
WEKA, MOA


 **Data streams**
MOA, Twitter


 **Imaging**
ImageJ, Java Advanced Imaging,
ImageMagick, Gnuplot

 **Scripting**
Groovy, Jython

 **Spreadsheets**
CSV, MS Excel, ODF

 **Graphics support**
BMP, JPG, PNG, TIF, PDF

 **Network**
http, (s)ftp, ssh, scp, email

 **Compression**
tar, zip, bzip2, gzip, lzf, lzma

Operators

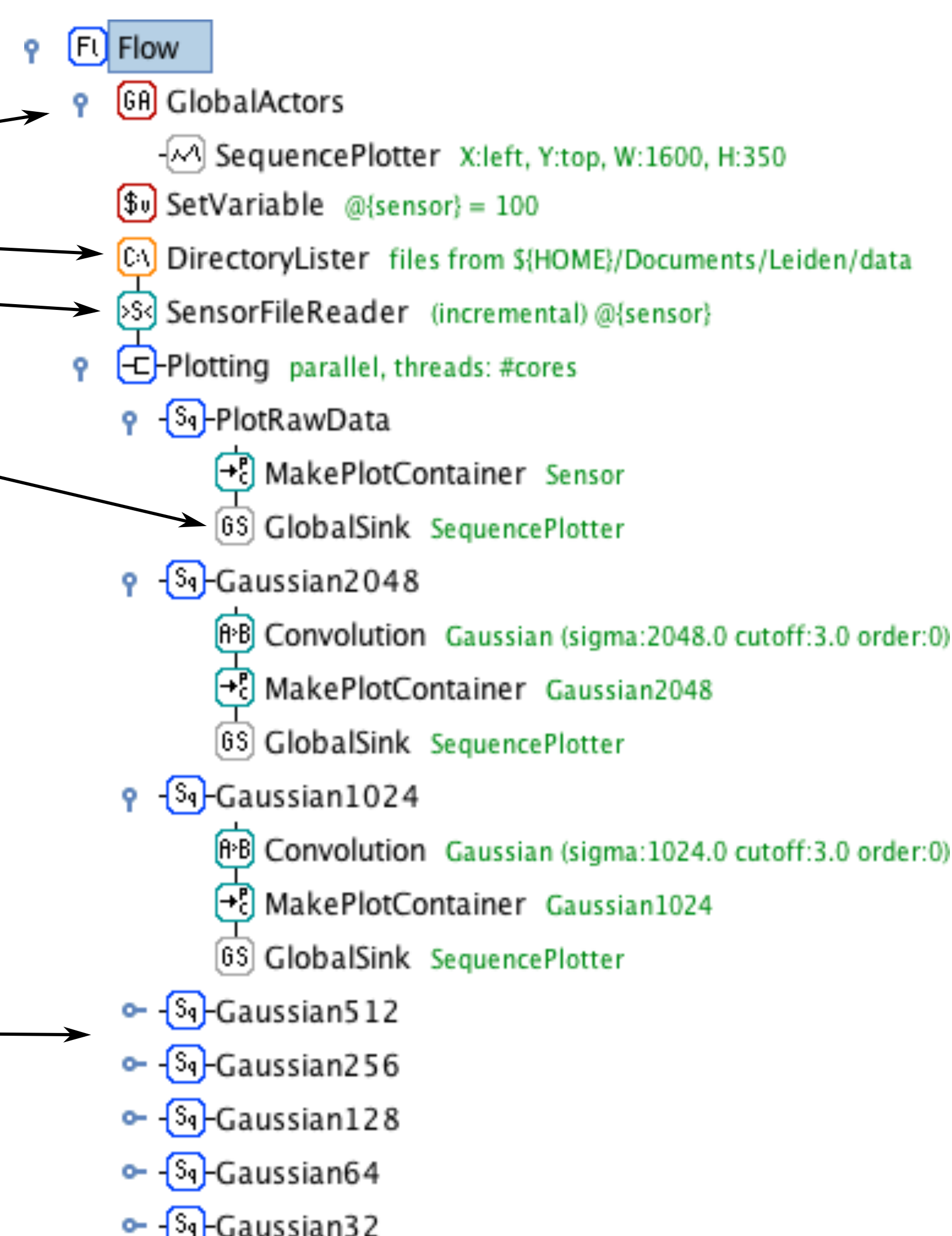
- **Standalone** (no input or output)
- **Source** (only output)
- **Transformer** (input and output)
- **Sink** (only input)

Tokens

Data is passed using tokens,
wrapping single Java Object
and Provenance information

Control actors

Determine how data flows,
e.g., Sequence, Branch, Tee,
Trigger, WhileLoop, If-Then-Else,
Switch, ...



Tree structure

- very compact layout
- can view very large flows
(by expanding only required sub-tree)
- no manual connecting of operators

N-to-M semantics

Tree cannot represent N-to-M,
but can be mitigated by:

- variables
- key-value pairs
- global actors

Interactivity

Certain actors can interact with user,
e.g., selecting file, prompting for
making a selection