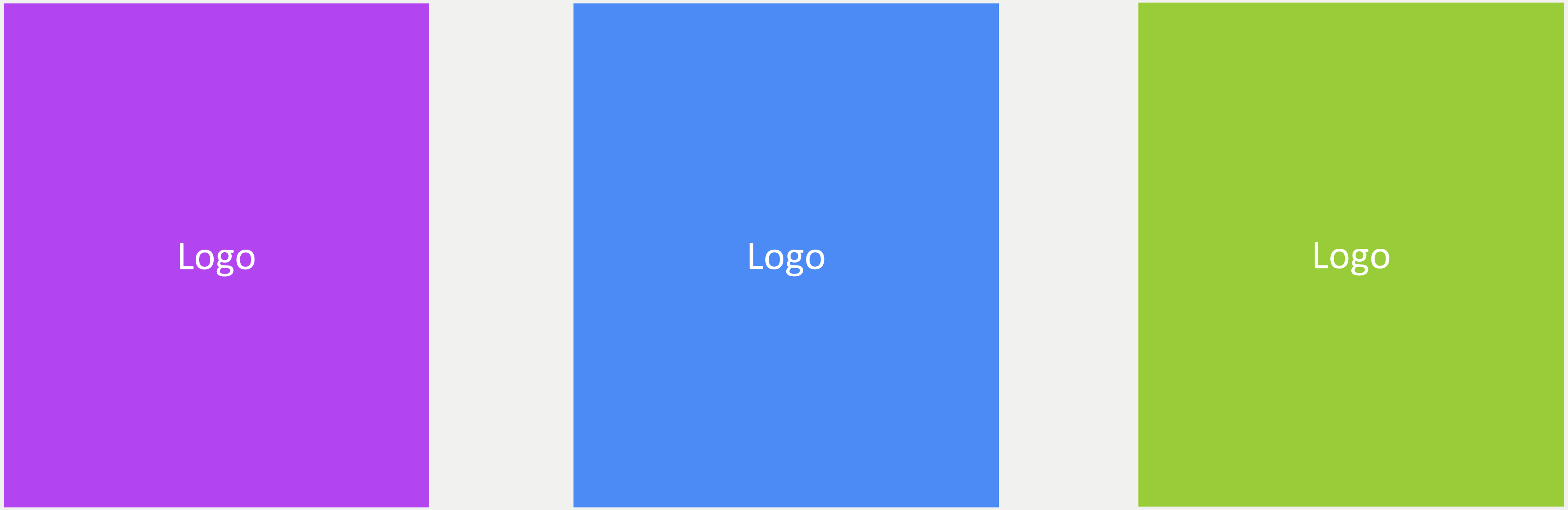
CONTENTS

- General
  - Tools & Plugins
  - Languages
  - Libraries
- Details
  - Implementation
  - Development
  - Architecture
- Live Preview
  - Algorithms and Features

Tools & Plugins



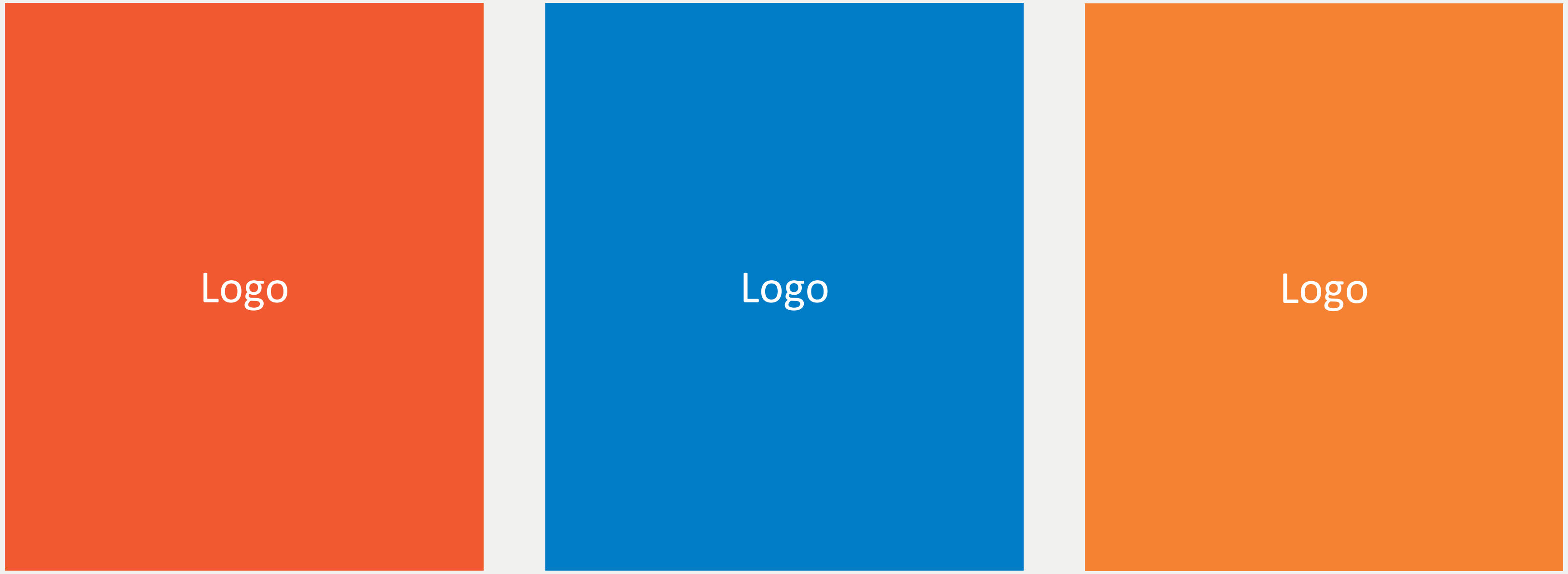
PhpStorm 2017.2.4

Chrome

JsTestDriver

Fig. 1.1: Logos of Chrome-Browser, IDE PhpStorm and Unit-Test -Environment JsTestDriver [P2, P3]

LANGUAGES



HTML 5

CSS 3

JavaScript
(ECMAScript 5.1)

Fig. 1.2: Logos of HTML 5, CSS 3 and JavaScript [P4]

LIBRARIES

jQuery

Knockout

namespace.js

FileSaver.js

jsPhyloSVG

Fig. 1.3: Logos of used libraries [P5-P7]

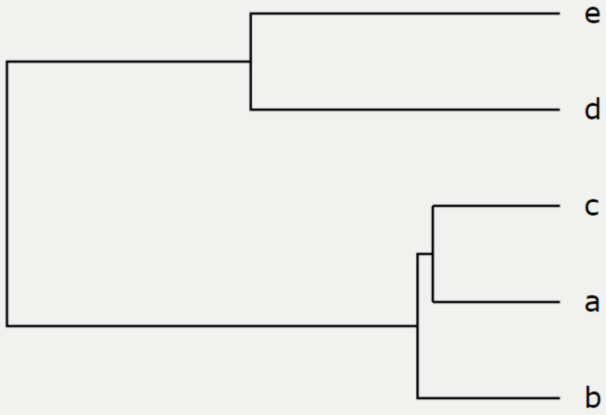


Fig. 1.4: Phylogenetic Tree created with jsPhyloSVG

Albert Ludwigs University Freiburg
Department of Computer Science
BIONFORMATICS GROUP FREIBURG

Details

IMPLEMENTATION, DEVELOPMENT
& ARCHITECTURE

IMPLEMENTATION

OBJECT-ORIENTED PROGRAMMING

```
(function() {
  // namespace name, "static" methods
  namespace("needlemanWunsch", NeedlemanWunsch);
  ...
  function NeedlemanWunsch() {
       // inheritance
       alignmentInstance = new bases.alignment.Alignment(this);

       this.setInput = alignmentInstance.setLinearAlignmentInput;
       this.compute = alignmentInstance.compute;
       this.getOutput = alignmentInstance.getOutput;

       // public methods (available through an instance)
       this.getSuperclass = getSuperclass;
  }
  ...
}());
```

Code 2.1: OOP-Simulation

IMPLEMENTATION

OBJECT-ORIENTED PROGRAMMING

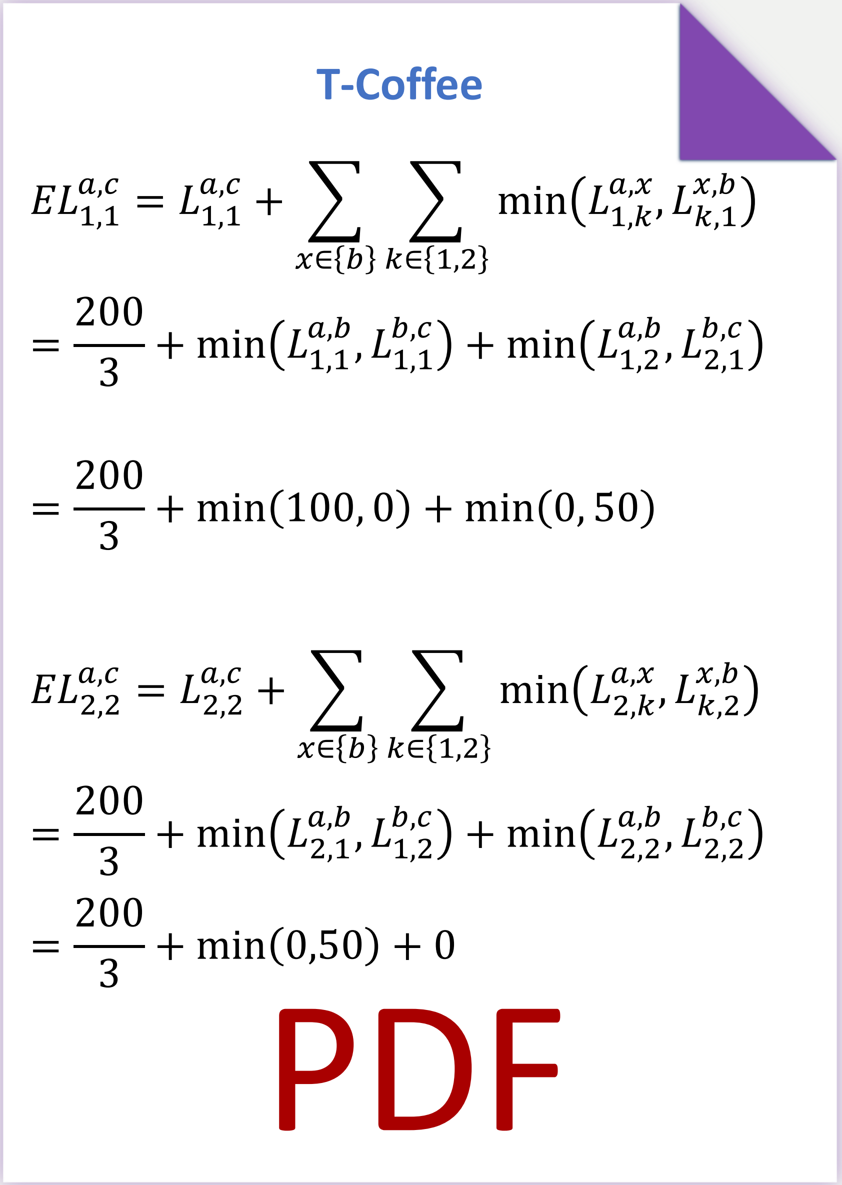
```
(function() {
  // namespace name, "static" methods
  namespace("needlemanWunsch", NeedlemanWunsch);

  function NeedlemanWunsch() {
       ...
       // public methods (available through an instance)
       this.getSuperclass = getSuperclass;
  }

  function a() {  // private, because not defined in constructor
       ...
  }
  ...
  function getSuperclass() {  // public, because in constructor
       return alignmentInstance;
  }
}());
```

Code 2.1: OOP-Simulation

Development

Extensively tested

- 52 Unit-Tests
  - step-by-step PDF-files
  - implementations with **JsTestDriver**



T-Coffee

$$EL^{a,c}_{1,1} = L^{a,c}_{1,1} + \sum_{x\in\{b\}} \sum_{k\in\{1,2\}} \min\left(L^{a,x}_{1,k}, L^{x,b}_{k,1}\right)$$

$$= \frac{200}{3} + \min\left(L^{a,b}_{1,1}, L^{b,c}_{1,1}\right) + \min\left(L^{a,b}_{1,2}, L^{b,c}_{2,1}\right)$$

$$= \frac{200}{3} + \min(100, 0) + \min(0, 50)$$

$$EL^{a,c}_{2,2} = L^{a,c}_{2,2} + \sum_{x\in\{b\}} \sum_{k\in\{1,2\}} \min\left(L^{a,x}_{2,k}, L^{x,b}_{k,2}\right)$$

$$= \frac{200}{3} + \min\left(L^{a,b}_{2,1}, L^{b,c}_{1,2}\right) + \min\left(L^{a,b}_{2,2}, L^{b,c}_{2,2}\right)$$

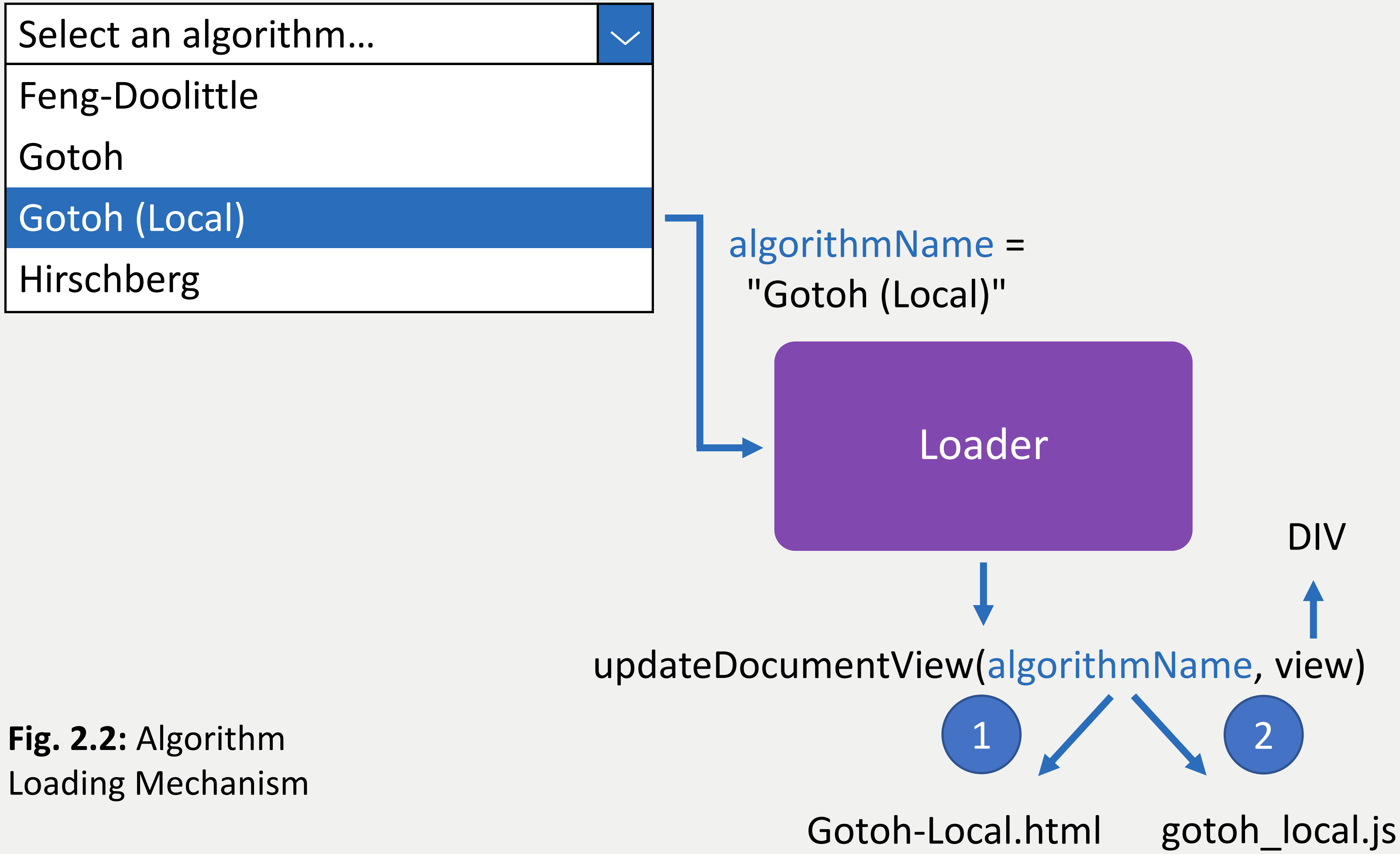
$$= \frac{200}{3} + \min(0{,}50) + 0$$

PDF

Fig. 2.1:
Excerpt from unit-test Notredame-Higgins-Heringa

ARCHITECTURE

LOADING



Select an algorithm…

Feng-Doolittle

Gotoh

Gotoh (Local)

Hirschberg

algorithmName = "Gotoh (Local)"

Loader

DIV

updateDocumentView(algorithmName, view)

1

2

Gotoh-Local.html

gotoh_local.js

Fig. 2.2: Algorithm Loading Mechanism

ARCHITECTURE

SEPARATION OF ALGORITHM AND INTERFACE LOGIC

D		A_1	A_2
	0	-2	-4
A_1	-2	1	-1
C_2	-4	-1	0
G_3	-6	-3	-2

Score: -2

Fig. 2.4: Highlighted Needleman-Wunsch matrix

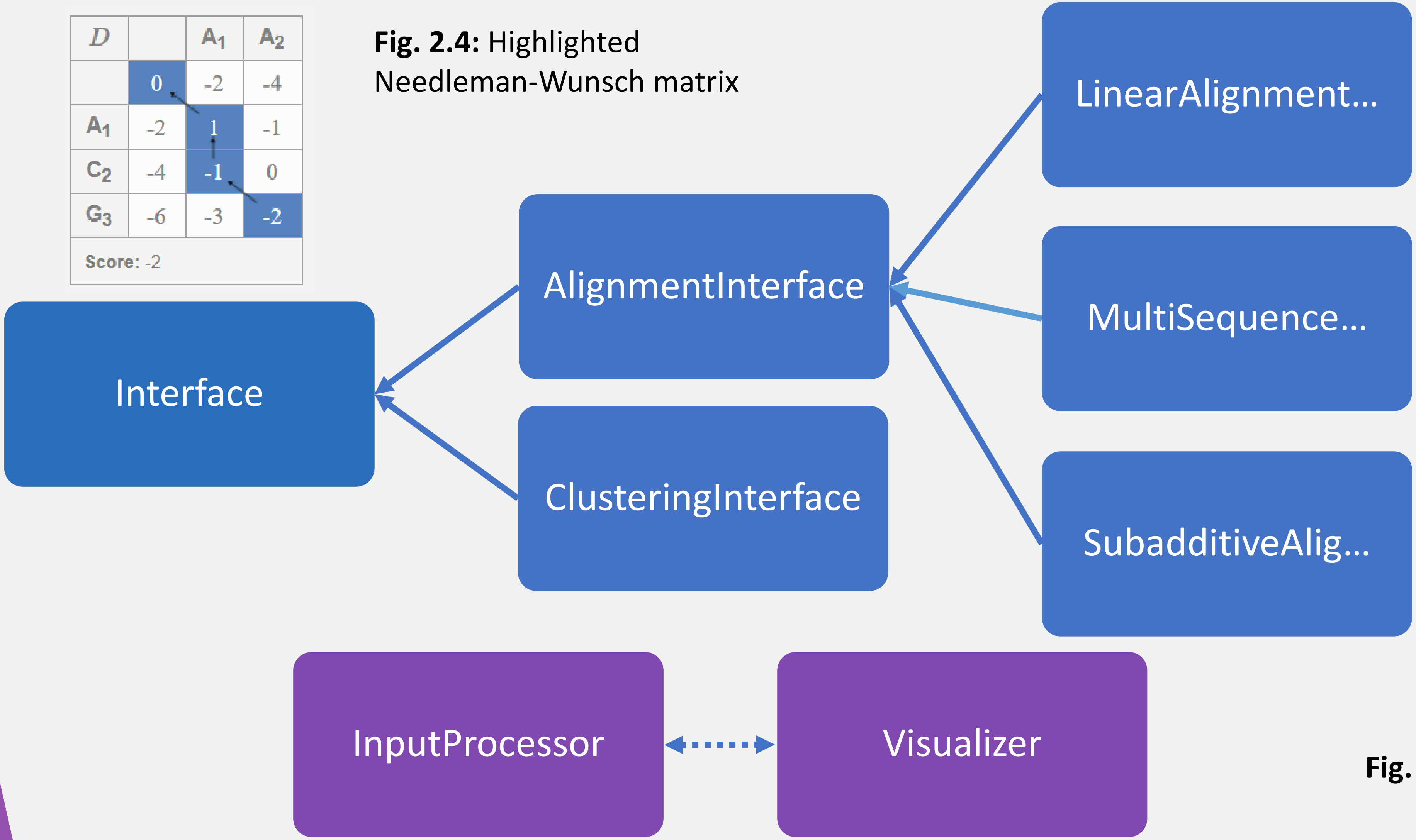
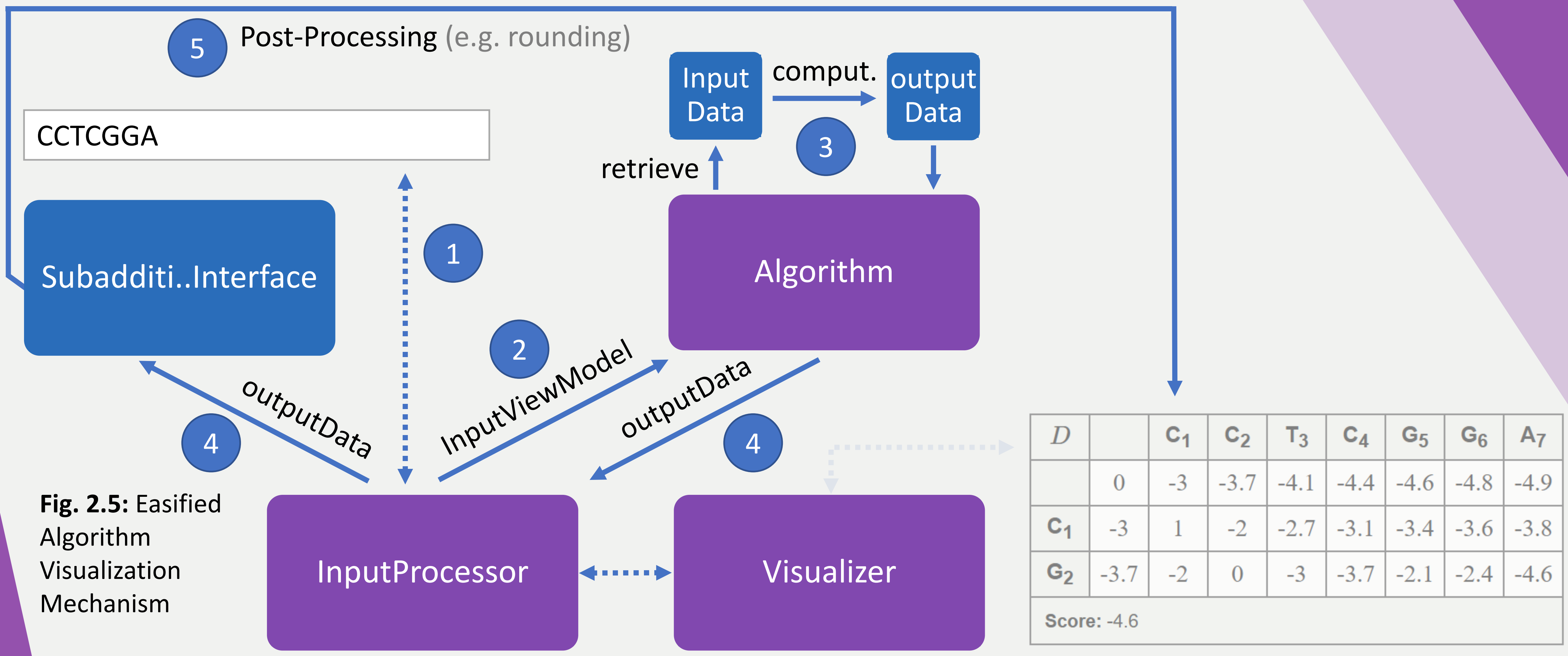


Fig. 2.3: Partial class diagram

Architecture

SEPARATION OF ALGORITHM AND INTERFACE LOGIC



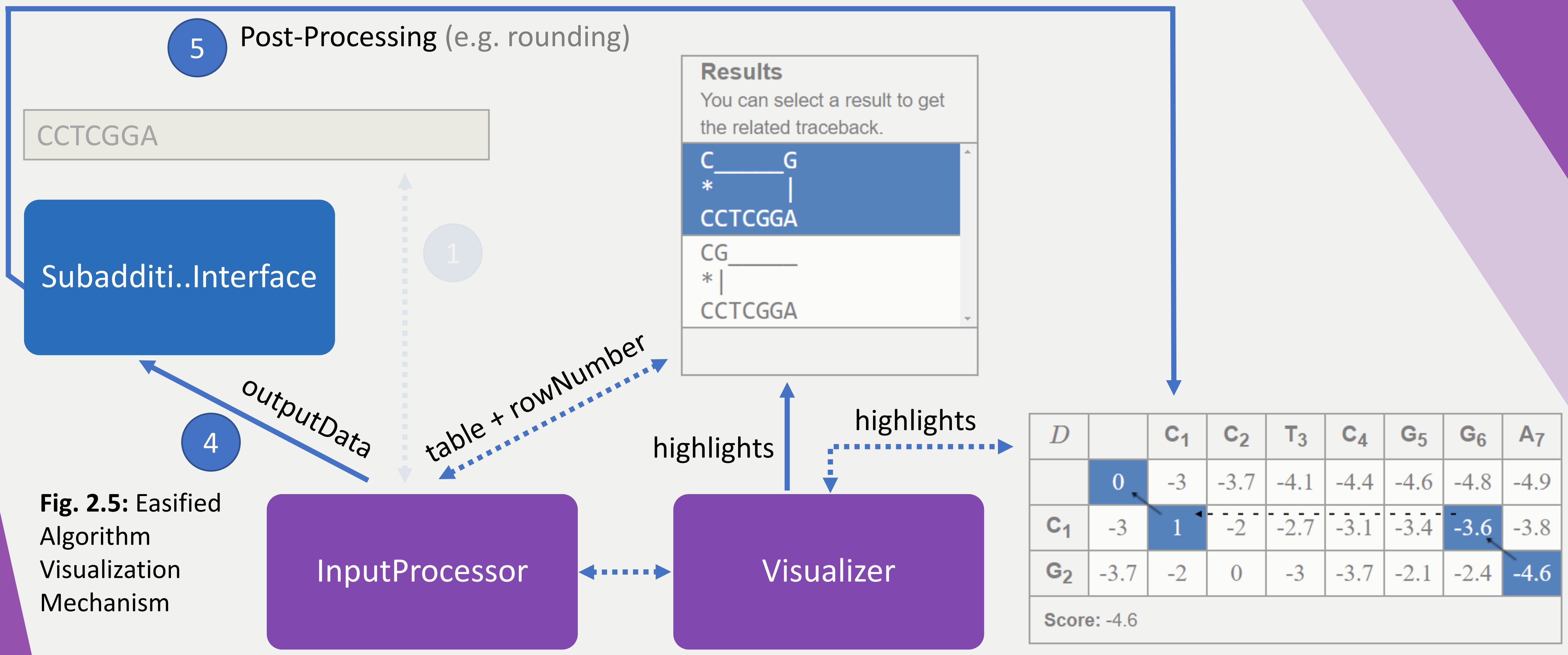
D		C_1	C_2	T_3	C_4	G_5	G_6	A_7
	0	-3	-3.7	-4.1	-4.4	-4.6	-4.8	-4.9
C_1	-3	1	-2	-2.7	-3.1	-3.4	-3.6	-3.8
G_2	-3.7	-2	0	-3	-3.7	-2.1	-2.4	-4.6

Score: -4.6

Fig. 2.5: Easified Algorithm Visualization Mechanism

ARCHITECTURE

SEPARATION OF ALGORITHM AND INTERFACE LOGIC



5 Post-Processing (e.g. rounding)

CCTCGGA

1

Subadditi..Interface

outputData

4

table + rowNumber

highlights

highlights

InputProcessor

Visualizer

Results

You can select a result to get the related traceback.

```
C_____G
*     |
CCTCGGA

CG_____
*|
CCTCGGA
```

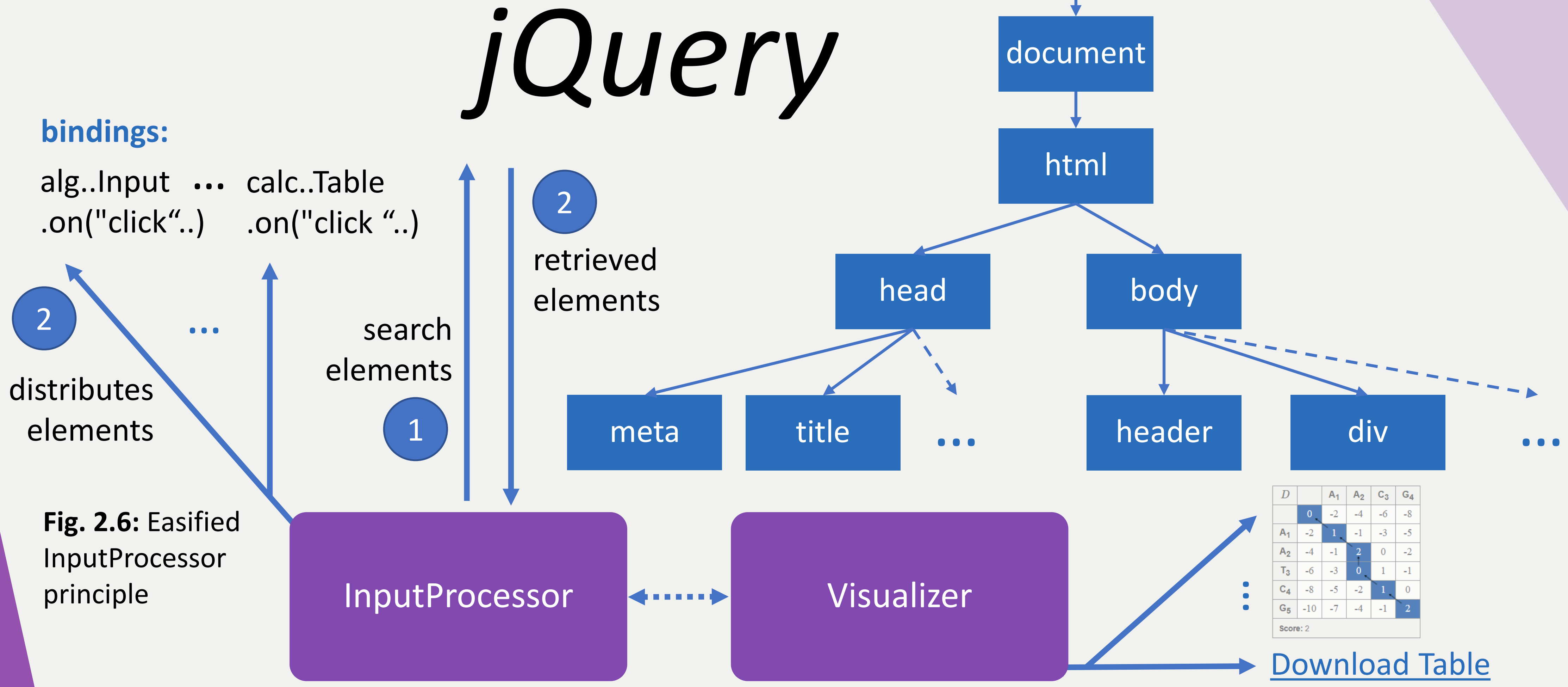
D		C_1	C_2	T_3	C_4	G_5	G_6	A_7
	0	-3	-3.7	-4.1	-4.4	-4.6	-4.8	-4.9
C_1	-3	1	-2	-2.7	-3.1	-3.4	-3.6	-3.8
G_2	-3.7	-2	0	-3	-3.7	-2.1	-2.4	-4.6

Score: -4.6

Fig. 2.5: Easified Algorithm Visualization Mechanism

ARCHITECTURE

ADVANTAGES – IDEA BEHIND



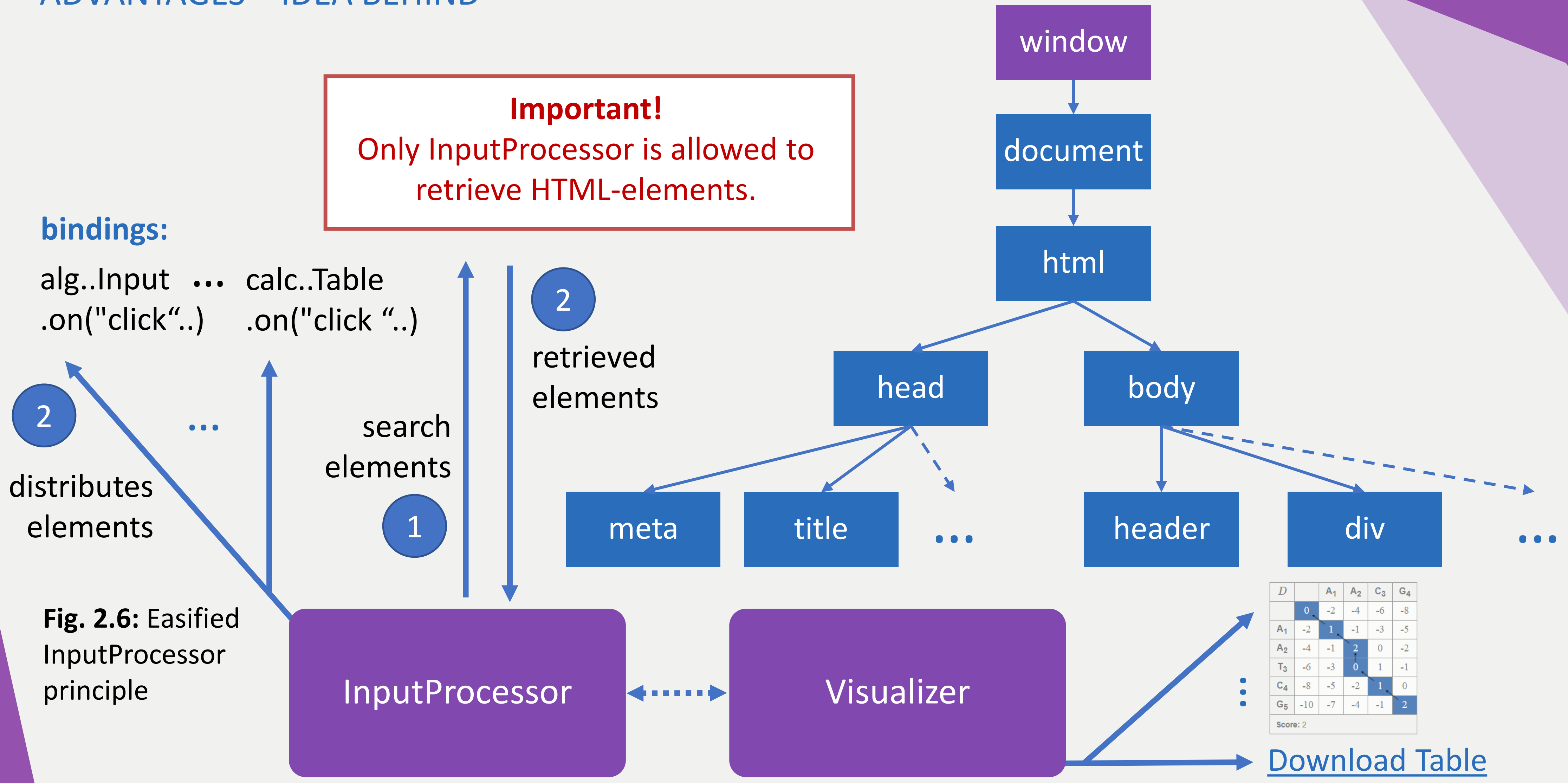
D		A$_1$	A$_2$	C$_3$	G$_4$
	0	-2	-4	-6	-8
A$_1$	-2	1	-1	-3	-5
A$_2$	-4	-1	2	0	-2
T$_3$	-6	-3	0	1	-1
C$_4$	-8	-5	-2	1	0
G$_5$	-10	-7	-4	-1	2

Score: 2

Fig. 2.6: Easified InputProcessor principle

ARCHITECTURE

ADVANTAGES – IDEA BEHIND



Important!
Only InputProcessor is allowed to retrieve HTML-elements.

bindings:

D		A_1	A_2	C_3	G_4
	0	-2	-4	-6	-8
A_1	-2	1	-1	-3	-5
A_2	-4	-1	2	0	-2
T_3	-6	-3	0	1	-1
C_4	-8	-5	-2	1	0
G_5	-10	-7	-4	-1	2

Score: 2

Fig. 2.6: Easified InputProcessor principle

SOURCES

- [1] 2017.12.09, jQuery Traversing,
  URL: https://www.w3schools.com/jquery/jquery_traversing.asp