Introduction to Conformance Checking

Fabrizio Maria Maggi
f.m.maggi@ut.ee
Process Analysis in the BPM Lifecycle
Process Mining

www.processmining.org
Process Mining
Replay: the Origin of Conformance Checking

Replay: the Origin of Conformance Checking

A B C E
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

B C E
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

C E
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

A C E
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

C E
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking

Diagram:

- A → + → B
- B → + → E
- A → X → C
- C → X → D
- D → + → E

End of diagram.
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Replay: the Origin of Conformance Checking
Dealing with Noise and Incompleteness

The **ideal process model** allows for the behavior coinciding with the frequent behavior seen when the process would be infinitely observed.

Mature process mining algorithms allow to **abstract** from infrequent behavior.
Four competing quality criteria

In general, the **quality** of a process mining result refers to **four quality dimensions**:

1. **Fitness**: the discovered model should allow for the behavior seen in the event log.
   - A model has a *perfect fitness* if all traces in the log can be replayed from the beginning to the end.
Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

1. **Fitness**
2. **Precision** (avoid underfitting): the discovered model should not allow for behavior completely unrelated to what was seen in the event log.
In general, the quality of a process mining result refers to four quality dimensions:

1. **Fitness**: 
2. **Precision** (avoid underfitting) 
3. **Generalization** (avoid overfitting): the discovered model should generalize the example behavior seen in the event log.
Four competing quality criteria

In general, the quality of a process mining result refers to four quality dimensions:

1. **Fitness**: the discovered model should allow for the behavior seen in the event log.
   - A model has a *perfect fitness* if all traces in the log can be replayed from the beginning to the end.

2. **Precision** (avoid underfitting): the discovered model should not allow for behavior completely unrelated to what was seen in the event log.

3. **Generalization** (avoid overfitting): the discovered model should generalize the example behavior seen in the event log.

4. **Simplicity**: the discovered model should be as simple as possible.
   - Occam’s Razor: The simplest model that can explain the behavior seen in the log is the best model.
Flower model (underfitting)

L = \{ <a, b, i, j, k, l>^{10}, <a, b, g, j, k, i, l>^{140}, <a, f, g, j, i, k>^{5}, <a, f, g, i, j, k, l>^{360} \}
Enumerating model (overfitting)

$L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}$
L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}
Computing fitness: basic approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360} \} \]
Computing fitness: basic approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360} \} \]
Computing fitness: basic approach

A B I J K L
Computing fitness: basic approach

A B I J K L
Computing fitness: basic approach

B I J K L
Computing fitness: basic approach

B I J K L
Computing fitness: basic approach

I J K L
Computing fitness: basic approach
Computing fitness: basic approach

I J K L

non-conformance
A “basic approach” to compute fitness is to **count the fraction of cases** that can be “parsed completely” (i.e., the proportion of cases corresponding to firing sequences leading from [start] to [end]).
A “basic approach” to compute fitness is to count the fraction of cases that can be “parsed completely” (i.e., the proportion of cases corresponding to firing sequences leading from [start] to [end]).

Fitness = 0.97
Computing fitness: Event-based approach

- In the simple fitness computation, we stopped replaying a **trace** once we encounter a problem and mark it as **non-fitting**.
- An event-based approach to calculate fitness consists of just continue replaying the trace on the model and:
  - record all situations where a transition is forced to fire without being enabled, i.e., **we count all missing tokens**.
  - record the **tokens that remain at the end**.
- Use of four counters:
  - $p =$ produced tokens
  - $c =$ consumed tokens
  - $m =$ missing tokens
  - $r =$ remaining tokens
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]
Computing fitness: Event-based approach

\[ L = \{ <a, b, i, j, k, l>^{10}, <a, b, g, j, k, i, l>^{140}, <a, f, g, j, i, k>^{5}, <a, f, g, i, j, k, l>^{360} \} \]
Computing fitness: Event-based approach

A B I J K L
Computing fitness: Event-based approach

A B I J K L
Computing fitness: Event-based approach

B I J K L

p = 1  c = 1  m = 0  r = 0
p = 1  c = 0  m = 0  r = 0
Computing fitness: Event-based approach
Computing fitness: Event-based approach
Computing fitness: Event-based approach

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]
Computing fitness: Event-based approach
Computing fitness: Event-based approach

I J K L

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]

\[ p = 1 \quad c = 1 \quad m = 0 \quad r = 0 \]
Computing fitness: Event-based approach
Computing fitness: Event-based approach
Computing fitness: Event-based approach
Computing fitness: Event-based approach
Computing fitness: Event-based approach
Computing fitness: Event-based approach

\[
\begin{align*}
p &= 1 \\
c &= 1 \\
m &= 0 \\
r &= 0
\end{align*}
\]
Computing fitness: Event-based approach

$L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\}$
Computing fitness: Event-based approach

L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}

\[ \text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right) \]

\( p = 12 \)
\( c = 12 \)
\( m = 1 \)
\( r = 1 \)
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360}\} \]

\[ p = 12 \]
\[ c = 12 \]
\[ m = 1 \]
\[ r = 1 \]

\[
\text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)
\]

Fitness = 0.9166
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]

\[
\text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right)
\]
Computing fitness: Event-based approach

L= \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[ p = 13 \]
\[ c = 13 \]
\[ m = 0 \]
\[ r = 0 \]

Fitness function:

\[ \text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right) \]

Fitness = 1
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[ \text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right) \]
Computing fitness: Event-based approach

$L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}$
Computing fitness: Event-based approach

$L = \{<a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360}\}$

$p = 12$
$c = 12$
$m = 1$
$r = 1$
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[ p = 12 \]
\[ c = 12 \]
\[ m = 1 \]
\[ r = 1 \]

Fitness = 0.9166
Computing fitness: Event-based approach

L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}
Computing fitness: Event-based approach

L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} 

\begin{align*}
\text{fitness}(\sigma, N) &= \frac{1}{2} \left(1 - \frac{m}{c}\right) + \frac{1}{2} \left(1 - \frac{r}{p}\right)
\end{align*}

\begin{align*}
p &= 13 \\
c &= 13 \\
m &= 0 \\
r &= 0
\end{align*}
Computing fitness: Event-based approach

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[ \text{fitness}(\sigma, N) = \frac{1}{2} \left( 1 - \frac{m}{c} \right) + \frac{1}{2} \left( 1 - \frac{r}{p} \right) \]

Fitness = 1
Computing fitness at log level

\[
L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \}
\]

\[
\text{fitness}(L, N) = \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}} \right) + \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}} \right)
\]
Computing fitness at log level

\[ L = \{<a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360}\} \]

\[ \text{fitness}(L,N) = \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times c_{N,\sigma}} \right) + \]

\[ \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_{N,\sigma}}{\sum_{\sigma \in L} L(\sigma) \times p_{N,\sigma}} \right) \]
Computing fitness at log level

\[ L = \{ \langle a, b, i, j, k, l \rangle^{10}, \langle a, b, g, j, i, k, l \rangle^{140}, \langle a, f, g, j, i, k \rangle^5, \langle a, f, g, i, j, k, l \rangle^{360} \} \]

\[ \begin{align*}
  p &= 12 & p &= 13 & p &= 12 & p &= 13 \\
  c &= 12 & c &= 13 & c &= 12 & c &= 13 \\
  m &= 1 & m &= 0 & m &= 1 & m &= 0 \\
  r &= 1 & r &= 0 & r &= 1 & r &= 0
\end{align*} \]

\[
\text{fitness}(L, N) = \frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times m_N, \sigma}{\sum_{\sigma \in L} L(\sigma) \times c_N, \sigma} \right) + \\
\frac{1}{2} \left( 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times r_N, \sigma}{\sum_{\sigma \in L} L(\sigma) \times p_N, \sigma} \right)
\]
Computing fitness at log level

\[ L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^{5}, <a,f,g,i,j,k,l>^{360} \} \]

\[
\begin{align*}
p &= 12 \\
c &= 12 \\
m &= 1 \\
r &= 1 \\
p &= 13 \\
c &= 13 \\
m &= 0 \\
r &= 0 \\
p &= 12 \\
c &= 12 \\
m &= 1 \\
r &= 1 \\
p &= 13 \\
c &= 13 \\
m &= 0 \\
r &= 0 \\
\text{Fitness} &= 0.997
\end{align*}
\]
Computing Precision

L = { <a, b, i, j, k, l>^{10}, <a, b, g, j, k, i, l>^{140}, <a, f, g, j, i, k>^{5}, <a, f, g, i, j, k, l>^{360}}
Computing Precision

$L = \{ <a,b,i,j,k,l>^{10}, <a,b,g,j,k,i,l>^{140}, <a,f,g,j,i,k>^5, <a,f,g,i,j,k,l>^{360}\}$

# distinct traces in the log conformant with the model (2) / # possible traces conformant with the model (6) = 0.33
A common approach to compute generalization is to check the quality of a discovered model.
A common approach to compute generalization in to check the quality of a discovered model
A common approach to compute generalization in to check the quality of a discovered model
A common approach to compute generalization in to check the quality of a discovered model
A common approach to compute generalization in to check the quality of a discovered model

Fitness
A common approach to compute generalization in to check the quality of a discovered model
A common approach to compute generalization in

to check the quality of a discovered model

Fitness
A common approach to compute generalization is to check the quality of a discovered model.
A common approach to compute generalization in order to check the quality of a discovered model.
Trace Alignment
Trace alignment

- Investigate relations between **moves in the log** and **moves in the model** to establish an alignment between a model $N$ and a trace $\sigma$.

<table>
<thead>
<tr>
<th>Move in the log only</th>
<th>Move in the model only</th>
</tr>
</thead>
<tbody>
<tr>
<td>a $\rightarrow$ d b e h</td>
<td>a c d $\rightarrow$ e h</td>
</tr>
</tbody>
</table>

**Alignment** of $\sigma = <adbeh>$

The top row corresponds to $\sigma$.

The bottom row corresponds to a path in the model (adjusted trace).

- If a move in the log cannot be mimicked by the model and vice-versa, such "no moves" are denoted by $\gg$ (and may have a **cost**).
Moves have costs

- **Standard cost function:**
  - $c(x, ») = 1$
  - $c(», y) = 1$
  - $c(x, y) = 0$, if $x = y$

**OPTIMAL ALIGNMENT**

alignment with minimum deviation cost

Any cost structure is possible!
Trace alignment

- Investigate relations between moves in the log and moves in the model to establish an alignment between a model N and a trace σ.

  Alignment of σ = <adbeh>

  Move in the log only
  Move in the model only

- If a move in the log cannot be mimicked by the model and vice-versa, such "no moves" are denoted by >> (and may have a cost).

  Is this alignment optimal?
Alignment-based fitness

- Calculate the fitness of the trace \(<a,d,b,e,h>\) with respect to the model \(N\)

\[
\begin{align*}
\text{Cost of worst-case alignment: } & 10 \\
\text{Cost of optimal alignment: } & 2 \\
\text{Fitness: } & 1 - \frac{2}{10} = 0.8
\end{align*}
\]

\[\text{SHORTEST PATH IN THE MODEL}\]
Optimal alignments

<abdeg>
Optimal alignments

<abdeg>
Optimal alignments

<abdeg>

optimal

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>d</th>
<th>e</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>»</td>
<td>d</td>
<td>e</td>
<td>g</td>
</tr>
</tbody>
</table>

1
Optimal alignments

<abdeg>

optimal

Worst-case
Optimal alignments

<abdeg>

optimal

Worst-case

| a | b | d | e | g | » | » | » | » | » | a | d | e | g |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | » | d | e | g | 1 |
| » | » | » | » | » | 9 |
Optimal alignments

<abdeg>

optimal

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>d</th>
<th>e</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>»</td>
<td>d</td>
<td>e</td>
<td>g</td>
</tr>
</tbody>
</table>

Worst-case

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>d</th>
<th>e</th>
<th>g</th>
<th>»</th>
<th>»</th>
<th>»</th>
<th>»</th>
</tr>
</thead>
<tbody>
<tr>
<td>»</td>
<td>»</td>
<td>»</td>
<td>»</td>
<td>»</td>
<td>a</td>
<td>d</td>
<td>e</td>
<td>g</td>
</tr>
</tbody>
</table>

Fitness: $1 - \frac{1}{9} = 0.89$
Fitness based on alignments

\[ \text{fitness}(\sigma, N) = 1 - \frac{\delta(\lambda_{opt}^N(\sigma))}{\delta(\lambda_{worst}^N(\sigma))} \]

In a worst-case alignment: (i) all events in trace \( \sigma \) are converted to log moves and (ii) a shortest path from an initial state to a final state of the model is added as a sequence of model moves.

Cost of the **optimal alignment** of the trace \( \sigma \)

Cost of the **worst-case alignment** where there are no synchronous moves and moves in model and log only.
Fitness for the entire log

\[ \text{fitness}(L, N) = 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times \delta(\lambda^N_{opt}(\sigma))}{\sum_{\sigma \in L} L(\sigma) \times \delta(\lambda^N_{worst}(\sigma))} \]

This is the sum of all costs when replaying the entire event log using optimal alignments.

Number of occurrences of a specific trace in the log (e.g., if a trace \( \sigma \) appears 200 times in the log, \( L(\sigma) \) will be equal to 200).

It is divided by the sum of the cost of all worst-case scenarios to obtain a normalized fitness value.
Direcly-Follows Map?

\[ L = \{ <a,b,c,d,e,f>^4, \]
\[ <a,g,h,d,f,e>^2, \]
\[ <a,b,c,d,f,e>^3, \]
\[ <a,g,h,d,e,f>^3 \} \]
Directly-Follows Map?

L = \{ <a,b,c,d,e,f>^{4},
       <a,g,h,d,f,e>^{2},
       <a,b,c,d,f,e>^{3},
       <a,g,h,d,e,f>^{3}\}
Directly-Follows Map?

\[ L = \{ <a,b,i,j,l>^{10}, <a,b,j,i,l>^{140}, <a,g,l>^{360} \} \]