

Code ▼

Process discovery was performed using bupaR, a suite of open-source R packages for business process data analysis.

1. Control flow
2. Time
3. Organisation
4. Outcome

Process discovery was performed using bupaR, a suite of open-source R packages for business process data analysis.

Hide

. _ _ . _ _ _ _ _ . _ _ _
 | _ \ | | | | | _ \ / \ \ \ / / | _ || _ \ / ||
 _ |
 | |) | | | | | |) | / ^ \ \ \ \ / / | | _ | |) | | (----`|
 | _
 | _ < | | | | | _ / / / _ \ \ \ / | _ | | / \ \ |
 _ |
 | |) | | `--' | | / _ _ \ \ / | | _ | | \ \ ----.----) | |
 | _
 | _ / \ _ / | _ / _ / \ _ \ _ / | _ || _ | `._ _ | _ / |
 _ |

```

— Attaching packages —
bupaverse 0.1.0 ✓ bupaR 0.5.3 ✓ processcheckR 0.1.4
✓ edeaR 0.9.4 ✓ processmapR 0.5.3
✓ eventdataR 0.3.1 — Conflicts —
bupaverse_conflicts() —
✖ bupaR::filter() masks stats::filter()
✖ processmapR::frequency() masks stats::frequency()
✖ edeaR::setdiff() masks base::setdiff()
✖ bupaR::timestamp() masks utils::timestamp()
✖ processcheckR::xor() masks base::xor()

```

Hide

```
event_log_df <- read.csv("initial_tasks.csv", stringsAsFactors = FALSE, sep = ",")
event_log_df <- event_log_df %>%
  convert_timestamps(columns = c("start", "complete"), format = ymd_hms) %>%
  activitylog(case_id = "taskId", activity_id = "action", resource_id = "actionBy", time
stamps = c("start", "complete"))
head(event_log_df)
```

```
# Log of 12 events consisting of:
2 traces
5 cases
6 instances of 2 activities
1 resource
Events occurred from 2021-12-01 04:45:26 until 2021-12-01 04:54:13
```

```
# Variables were mapped as follows:
Case identifier:      taskId
Activity identifier:  action
Resource identifier:  actionBy
Timestamps:          start, complete
```

X	taskId	action	actionBy	start	compl
<int>	<chr>	<chr>	<chr>	<S3: POSIXct>	<S3: POSIXct>
0	11875_33	LOCKED_FOR_MAPPING	fajarramadhana	2021-12-01 04:45:26	2021-12-01 04:46:13
1	11875_72	LOCKED_FOR_MAPPING	fajarramadhana	2021-12-01 04:46:37	2021-12-01 04:47:24
2	11875_85	LOCKED_FOR_MAPPING	fajarramadhana	2021-12-01 04:47:24	2021-12-01 04:48:11
3	11875_42	LOCKED_FOR_MAPPING	fajarramadhana	2021-12-01 04:49:53	2021-12-01 04:50:40
4	11875_42	MAPPED	fajarramadhana	2021-12-01 04:53:23	2021-12-01 04:53:23
5	11875_29	LOCKED_FOR_MAPPING	fajarramadhana	2021-12-01 04:53:55	2021-12-01 04:54:13

6 rows

Control flow

Absolute frequency of activities in the eventlog.

Hide

```
event_log_df %>% activity_frequency("activity")
```

action	absolute	relative
<chr>	<int>	<dbl>
LOCKED_FOR_MAPPING	406324	0.305448204
LOCKED_FOR_VALIDATION	308244	0.231717979

action <chr>	absolute <int>	relative <dbl>
MAPPED	284214	0.213653773
VALIDATED	284035	0.213519212
AUTO_UNLOCKED_FOR_MAPPING	20070	0.015087333
INVALIDATED	9429	0.007088115
SPLIT	7680	0.005773329
BADIMAGERY	4502	0.003384314
AUTO_UNLOCKED_FOR_VALIDATION	2956	0.002222130
EXTENDED_FOR_MAPPING	2801	0.002105611
1-10 of 10 rows		

Activity presence shows in what percentage of cases an activity is present.

[Hide](#)

```
event_log_df %>% activity_presence()
```

action <chr>	absolute <int>	relative <dbl>
LOCKED_FOR_MAPPING	274340	0.987740553
LOCKED_FOR_VALIDATION	271218	0.976500027
MAPPED	271187	0.976388414
VALIDATED	270091	0.972442348
AUTO_UNLOCKED_FOR_MAPPING	14733	0.053045059
INVALIDATED	8201	0.029527084
SPLIT	7680	0.027651263
BADIMAGERY	4124	0.014848152
AUTO_UNLOCKED_FOR_VALIDATION	2796	0.010066788
EXTENDED_FOR_MAPPING	766	0.002757925
1-10 of 10 rows		

The start of cases can be described using the start_activities function.

[Hide](#)

```
event_log_df %>% start_activities("activity")
```

action <chr>	absolute <int>	relative <dbl>	cum_sum <dbl>
LOCKED_FOR_MAPPING	265857	9.571981e-01	0.9571981
AUTO_UNLOCKED_FOR_MAPPING	8598	3.095645e-02	0.9881546
SPLIT	3095	1.114331e-02	0.9992979
MAPPED	79	2.844336e-04	0.9995824
BADIMAGERY	60	2.160255e-04	0.9997984
LOCKED_FOR_VALIDATION	48	1.728204e-04	0.9999712
VALIDATED	4	1.440170e-05	0.9999856
INVALIDATED	3	1.080127e-05	0.9999964
EXTENDED_FOR_MAPPING	1	3.600425e-06	1.0000000
9 rows			

The end_activities function describes the end of cases

[Hide](#)

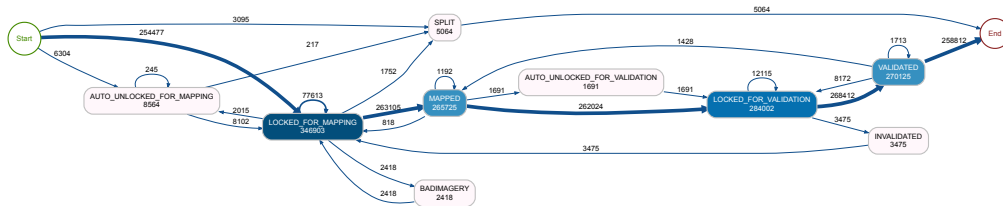
```
event_log_df %>% end_activities("activity")
```

action <chr>	absolute <int>	relative <dbl>	cum_sum <dbl>
VALIDATED	269844	9.715530e-01	0.9715530
SPLIT	7642	2.751445e-02	0.9990675
BADIMAGERY	185	6.660786e-04	0.9997336
AUTO_UNLOCKED_FOR_MAPPING	38	1.368161e-04	0.9998704
MAPPED	30	1.080127e-04	0.9999784
LOCKED_FOR_VALIDATION	4	1.440170e-05	0.9999928
AUTO_UNLOCKED_FOR_VALIDATION	2	7.200850e-06	1.0000000
7 rows			

In the frequency process map, nodes represent the absolute number of activity instance executions and edges represent the absolute number of times source and target activities were executed directly following each other. To provide a clear process map, the event log was previously filter using filter_trace_frequency(). Setting percentage = 0.95 selects at least 95% of the cases, starting with those that have the highest frequency.

[Hide](#)

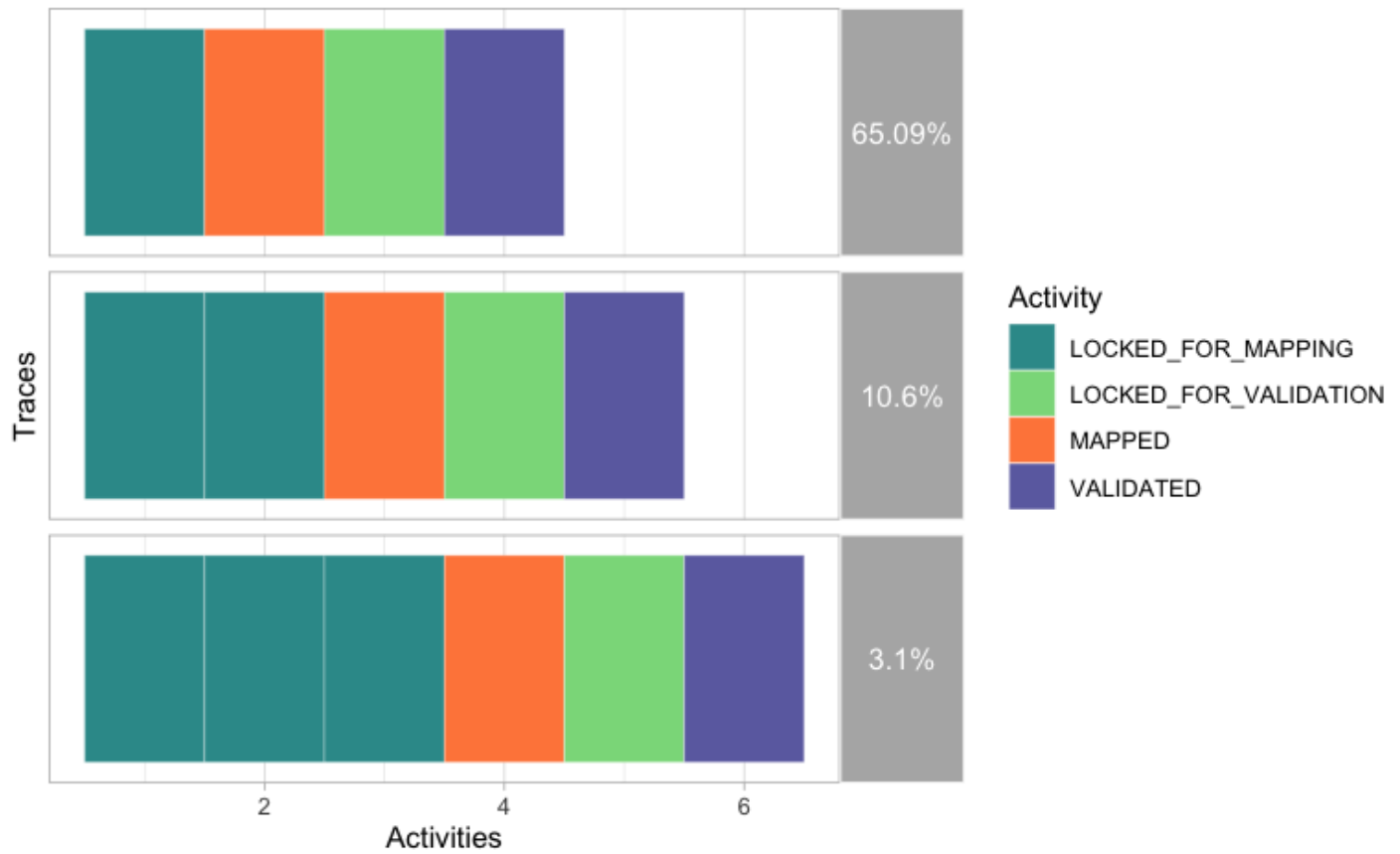
```
tmp <- event_log_df %>% filter_trace_frequency(percentage = 0.95)
tmp %>% process_map(frequency("absolute"))
```



`trace_explorer()` with coverage argument `n_traces = 3` shows the 3 most frequent in the event log.

Hide

```
event_log_df %>%
  trace_explorer(n_traces = 3, show_labels = FALSE, coverage_labels = c("relative"))
```

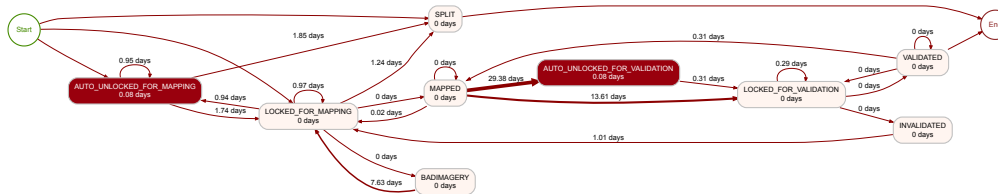


Time

In the temporal process map, the value of nodes and edges represent the median duration in days of activities and waiting times.

[Hide](#)

```
tmp %>% process_map(performance(median, "days"))
```



Timestamps are properly formatted for time calculations

Hide

```

event_log_df <- event_log_df %>% convert_timestamps(columns = c("start", "complete"), fo
rmat = ymd_hms)
event_log_df$time_diff=as.numeric(event_log_df$complete-event_log_df$start)
task_duration <- event_log_df %>% group_by(taskId) %>% summarise(min = min(start), max
= max(complete))
task_duration$duration=as.numeric(task_duration$max-task_duration$min)
task_duration <- task_duration[,c("taskId","duration")]
  
```

We calculated the relative time devoted for mapping activities ('LOCKED_FOR_MAPPING', 'AUTO_UNLOCKED_FOR_MAPPING') and validation activities ('LOCKED_FOR_VALIDATION', 'AUTO_UNLOCKED_FOR_VALIDATION') per case expressed as percentage of total case duration. The remaining time is considered idle.

Hide

```
mapping_duration <- filter(event_log_df, action == 'LOCKED_FOR_MAPPING' | action == 'AUTO_UNLOCKED_FOR_MAPPING') %>% group_by(taskId) %>% summarise(mapping = sum(time_diff))
validation_duration <- filter(event_log_df, action == 'LOCKED_FOR_VALIDATION' | action == 'AUTO_UNLOCKED_FOR_VALIDATION') %>% group_by(taskId) %>% summarise(validation = sum(time_diff))
durations <- merge(x = merge(x = task_duration, y = mapping_duration, by = "taskId", all.x = TRUE), y = validation_duration, by = "taskId", all.x = TRUE)
durations[is.na(durations)] <- 0
durations$mapping_per=durations$mapping/durations$duration*100
durations$validation_per=durations$validation/durations$duration*100
durations$service=durations$mapping_per+durations$validation_per
durations$idle_per=100-durations$mapping_per-durations$validation_per
```

Median % of iddle time per case

Hide

```
median(durations$idle_per, na.rm = TRUE )
```

```
[1] 99.93861
```

Median % of mapping and validation time per case

Hide

```
median(durations$service, na.rm = TRUE )
```

```
[1] 0.06139425
```

Median % of mapping time per case

Hide

```
median(durations$mapping_per, na.rm = TRUE )
```

```
[1] 0.01586283
```

Median % of validation time per case

Hide

```
median(durations$validation_per, na.rm = TRUE )
```

```
[1] 0.02293203
```

Organisation

Outcome

