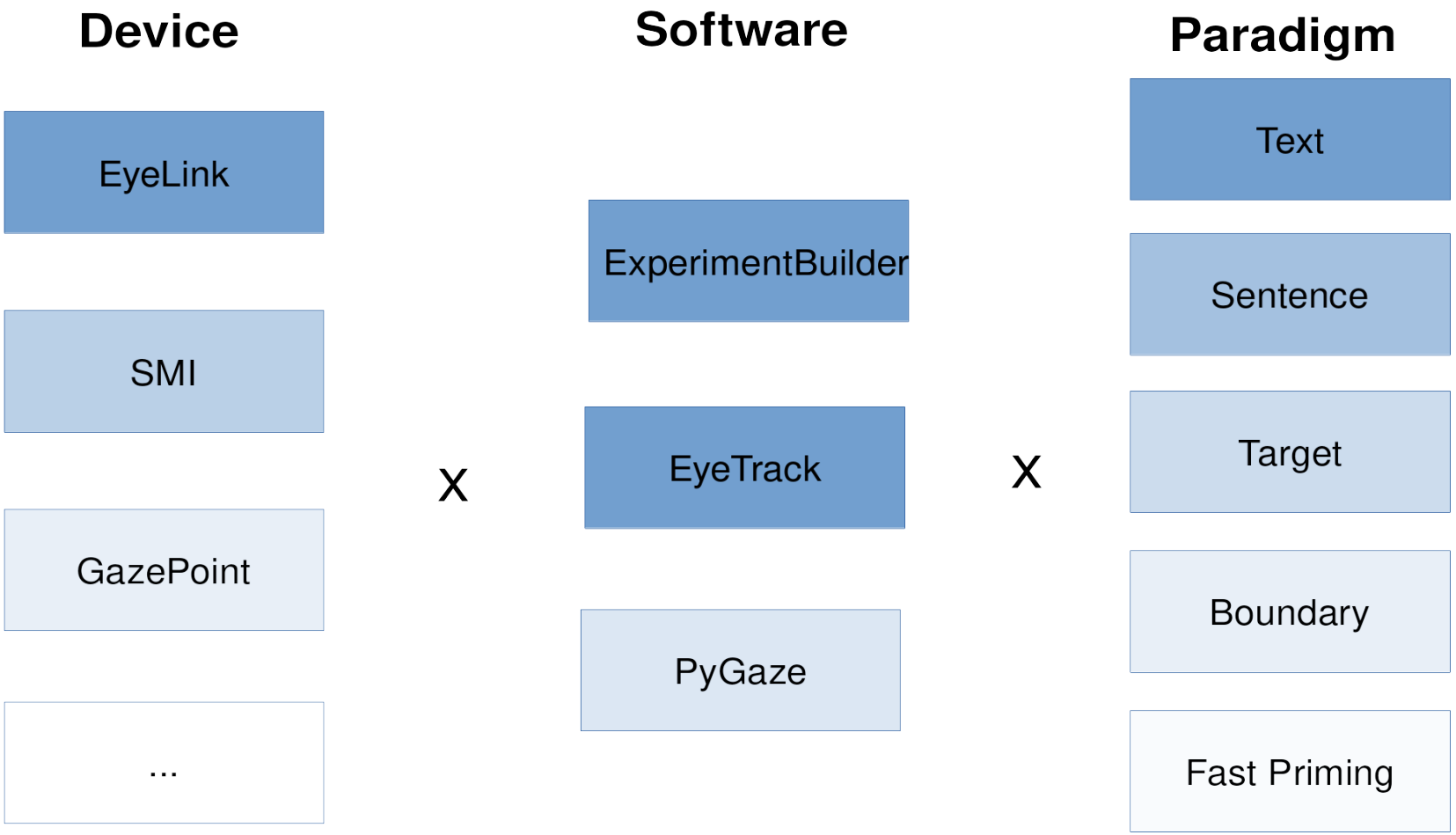
popEye -
An R package to analyse eye movement data from reading experiments

Sascha Schroeder

University of Göttingen
sascha.schroeder@psych.uni-goettingen.de

August 19, 2019

Why popEye?


Device
EyeLink
SMI
GazePoint
...
X
Software
ExperimentBuilder
EyeTrack
PyGaze
X
Paradigm
Text
Sentence
Target
Boundary
Fast Priming

Why popEye?

- 1. Unified data analysis across different hardware and software platforms.
- 2. Focus on *reading*: Hierarchical structure of language can be exploited during analysis.
- 3. Full control over pre-processing and analysis, better visualization and cleaning options.

Why popEye?

- 1. Unified data analysis across different hardware and software platforms.
- 2. Focus on *reading*: Hierarchical structure of language can be exploited during analysis.
- 3. Full control over pre-processing and analysis, better visualization and cleaning options.

Why popEye?

- 1. Unified data analysis across different hardware and software platforms.
- 2. Focus on *reading*: Hierarchical structure of language can be exploited during analysis.
- 3. Full control over pre-processing and analysis, better visualization and cleaning options.

Overview

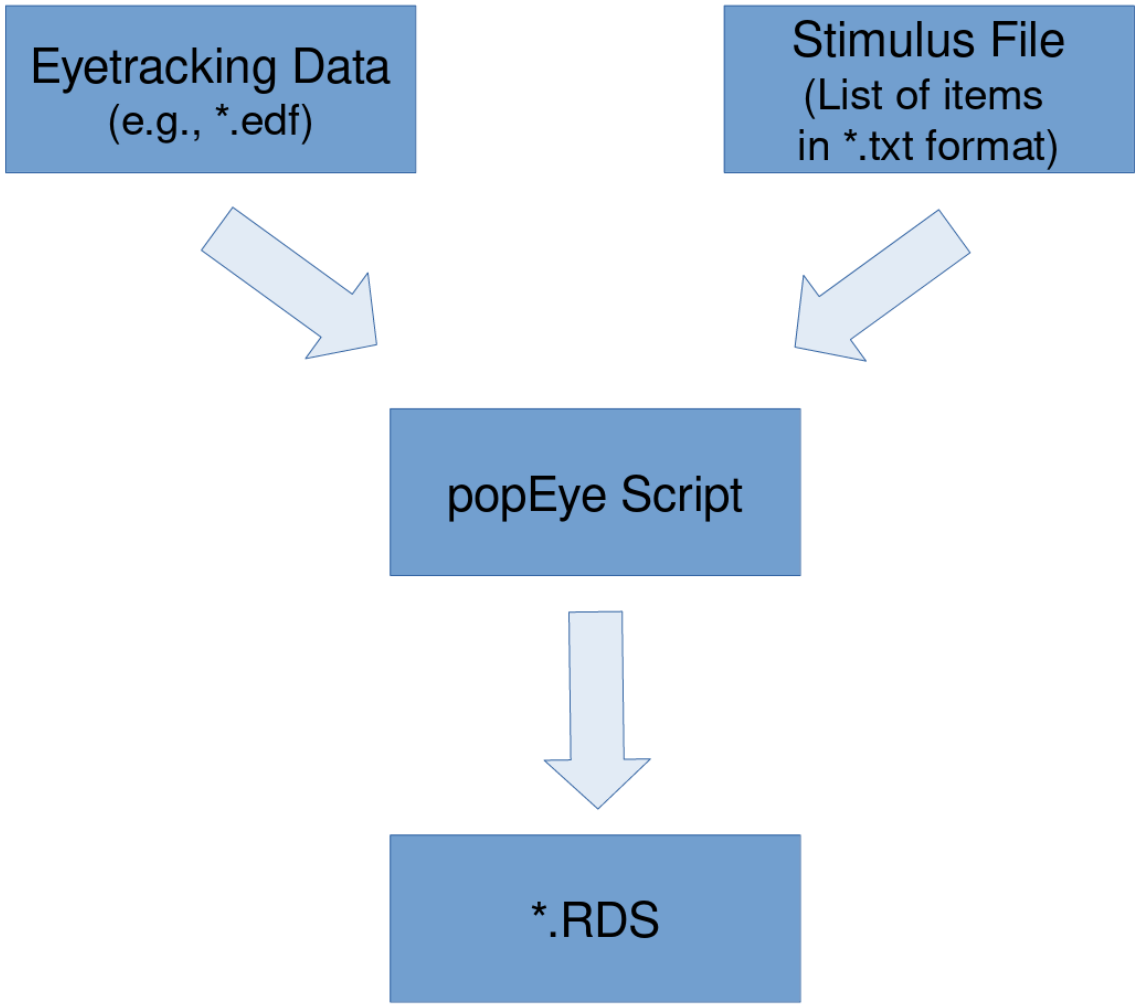
Preprocessing

- Read Data
- Parse Fixations
- Assign Stimuli
- Compute Measures
- Aggregate

Analysis

- Reports
- Plots
- Cleaning

Preprocessing



Preprocessing

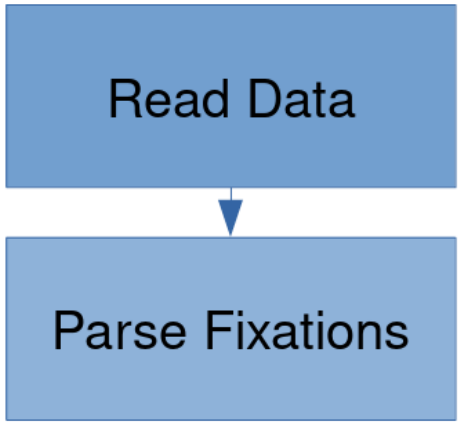
A	B	C	D
id	cond	preview	target
1	identreal	Der alte Mann stieg *ängstlich die Treppe hinab.	Der alte Mann stieg *ängstlich die Treppe hinab.
1	identpseudo	Der alte Mann stieg *ängsthaft die Treppe hinab.	Der alte Mann stieg *ängstlich die Treppe hinab.
1	identnon	Der alte Mann stieg *ängstnauf die Treppe hinab.	Der alte Mann stieg *ängstlich die Treppe hinab.
1	diffreal	Der alte Mann stieg *honiglich die Treppe hinab.	Der alte Mann stieg *ängstlich die Treppe hinab.
1	diffpseudo	Der alte Mann stieg *honighaft die Treppe hinab.	Der alte Mann stieg *ängstlich die Treppe hinab.
1	diffnon	Der alte Mann stieg *honignauf die Treppe hinab.	Der alte Mann stieg *ängstlich die Treppe hinab.
2	identreal	Ihr ganzer Körper wirkte *athletisch und durchtrainiert.	Ihr ganzer Körper wirkte *athletisch und durchtrainiert.
2	identpseudo	Ihr ganzer Körper wirkte *athlethaft und durchtrainiert.	Ihr ganzer Körper wirkte *athletisch und durchtrainiert.
2	identnon	Ihr ganzer Körper wirkte *athletpern und durchtrainiert.	Ihr ganzer Körper wirkte *athletisch und durchtrainiert.
2	diffreal	Ihr ganzer Körper wirkte *schaumisch und durchtrainiert.	Ihr ganzer Körper wirkte *athletisch und durchtrainiert.
2	diffpseudo	Ihr ganzer Körper wirkte *schaumhaft und durchtrainiert.	Ihr ganzer Körper wirkte *athletisch und durchtrainiert.
2	diffnon	Ihr ganzer Körper wirkte *schaumpern und durchtrainiert.	Ihr ganzer Körper wirkte *athletisch und durchtrainiert.
3	identreal	Sie hörten das kleine *Bächlein friedlich plätschern.	Sie hörten das kleine *Bächlein friedlich plätschern.
3	identpseudo	Sie hörten das kleine *Bächchen friedlich plätschern.	Sie hörten das kleine *Bächlein friedlich plätschern.
3	identnon	Sie hörten das kleine *Bächnauf friedlich plätschern.	Sie hörten das kleine *Bächlein friedlich plätschern.
3	diffreal	Sie hörten das kleine *Wirtlein friedlich plätschern.	Sie hörten das kleine *Bächlein friedlich plätschern.
3	diffpseudo	Sie hörten das kleine *Wirtchen friedlich plätschern.	Sie hörten das kleine *Bächlein friedlich plätschern.
3	diffnon	Sie hörten das kleine *Wirtnauf friedlich plätschern.	Sie hörten das kleine *Bächlein friedlich plätschern.

Preprocessing

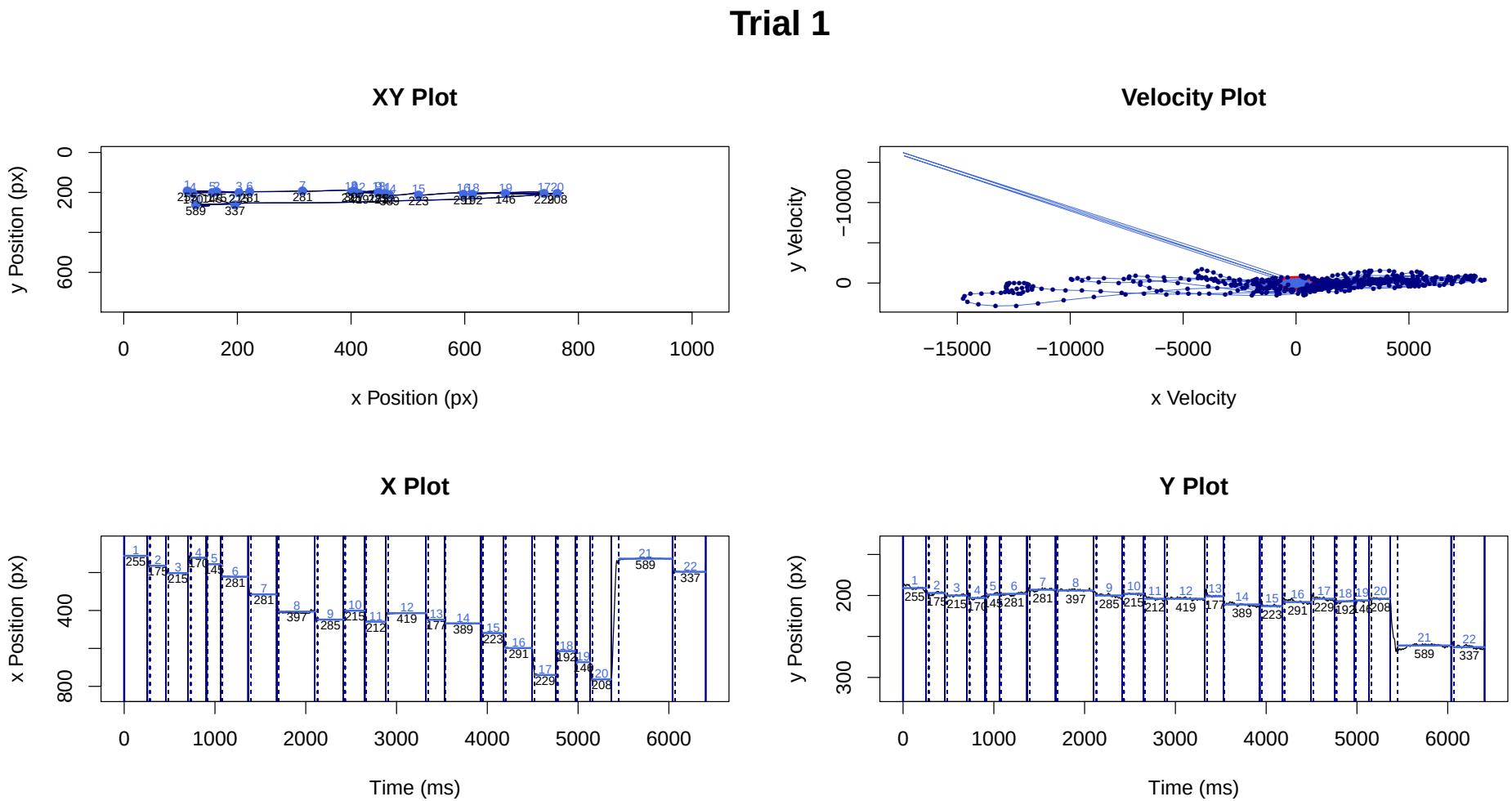
```r

# load package
library(popEye)

# load stimulus file
stimfile <- read.table("/data/gwdg/4_Research/MultiLing/texts/L2_Excelsheet_2019_03_25_tm.csv",
                       header = T, sep = ",", as.is = T, fileEncoding = "UTF-8", quote = "\"")

# call popEye
exp <- popEye(
  datpath = "/data/gwdg/4_Research/MultiLing/pilot/L2_2019_04_08_deploy/"
  , tracker.software = "EB"
  , type = "sentence"
  , tracker.results = F
  , message.stop = "blank_screen"
  , variable.id = "number"

  , stimulus.file = stimfile
  , stimulus.id = "Number"
  , stimulus.text = "Paragraph"

  , display.marginLeft = 100
  , display.marginTop = 80
  , display.marginRight = 100

  , font.name = "Consolas"
  , font.size = 20
  , font.spacing = 1

  , analysis.lineMethod = "chain"
  , clean.outlier = T

  , outpath = "/data/gwdg/4_Research/MultiLing/pilot/"
  , outname = "MultiLing_L2"
)

```

Preprocessing: Read Data

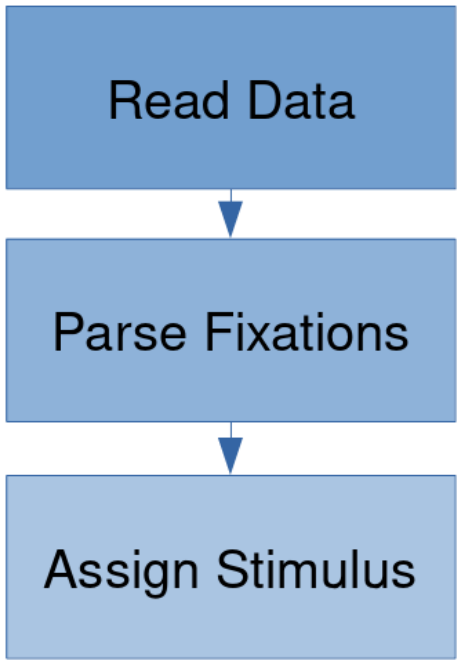
Preprocessing: Parse Fixations



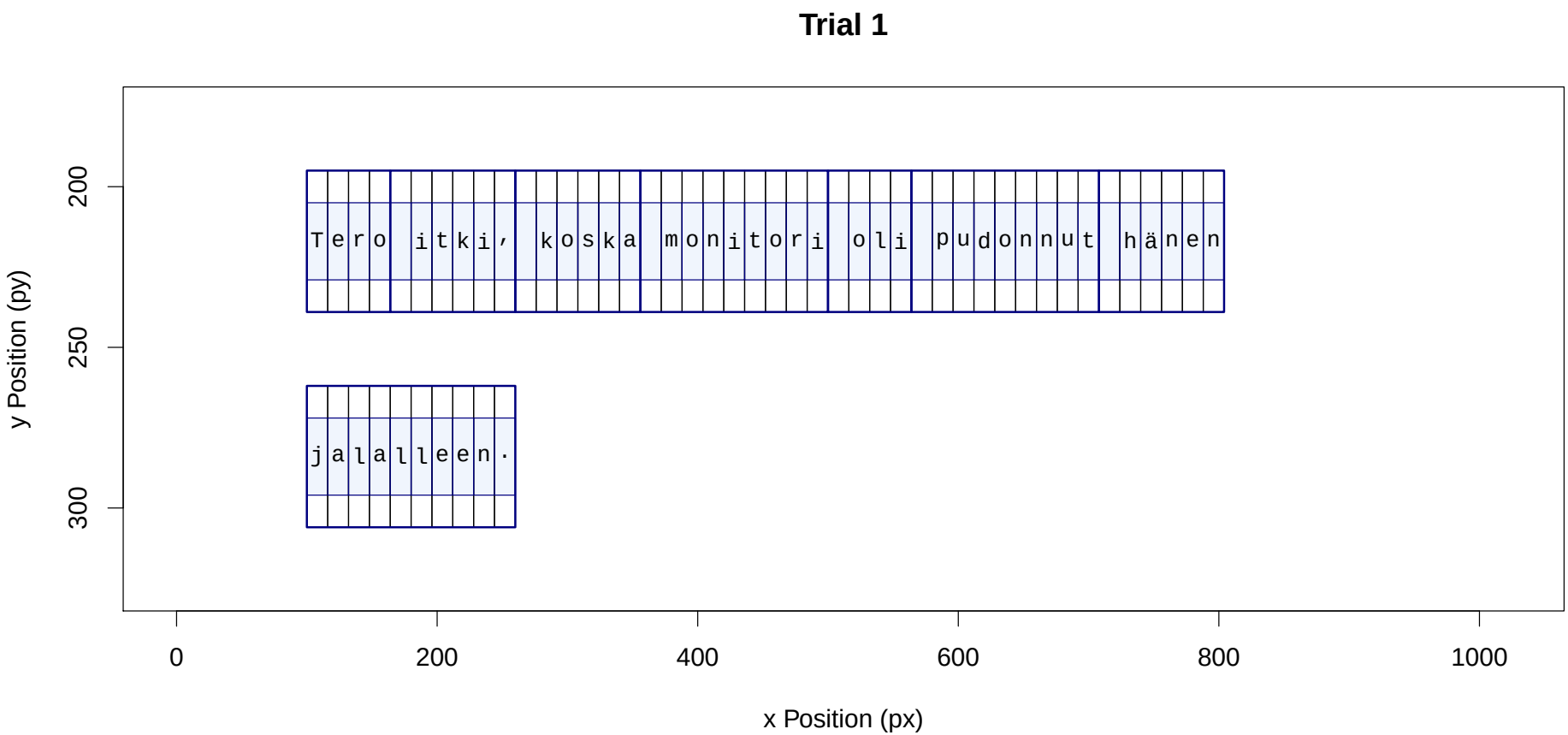
Preprocessing: Parse Fixations



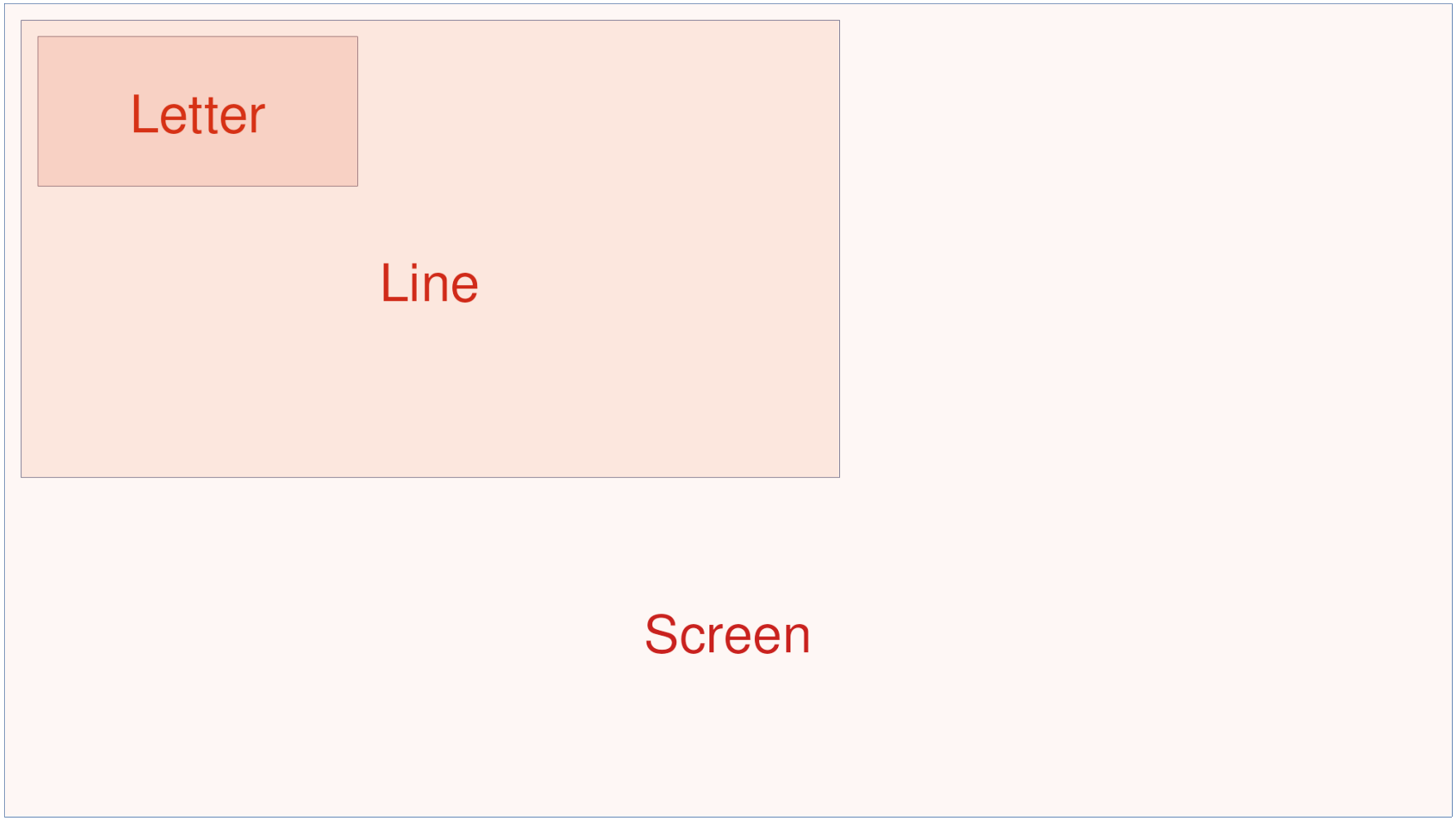
Preprocessing: Assign Stimuli



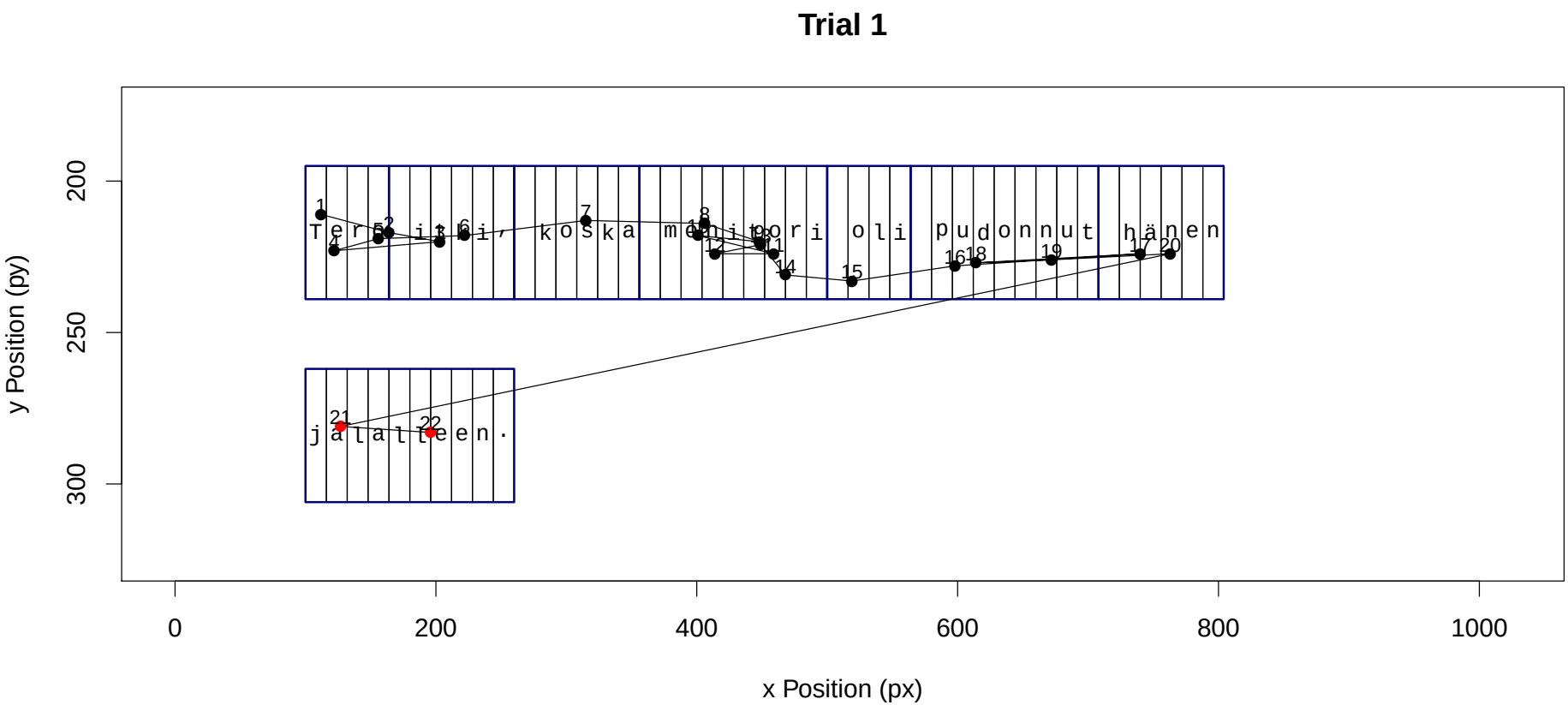
Preprocessing: Assign Stimuli



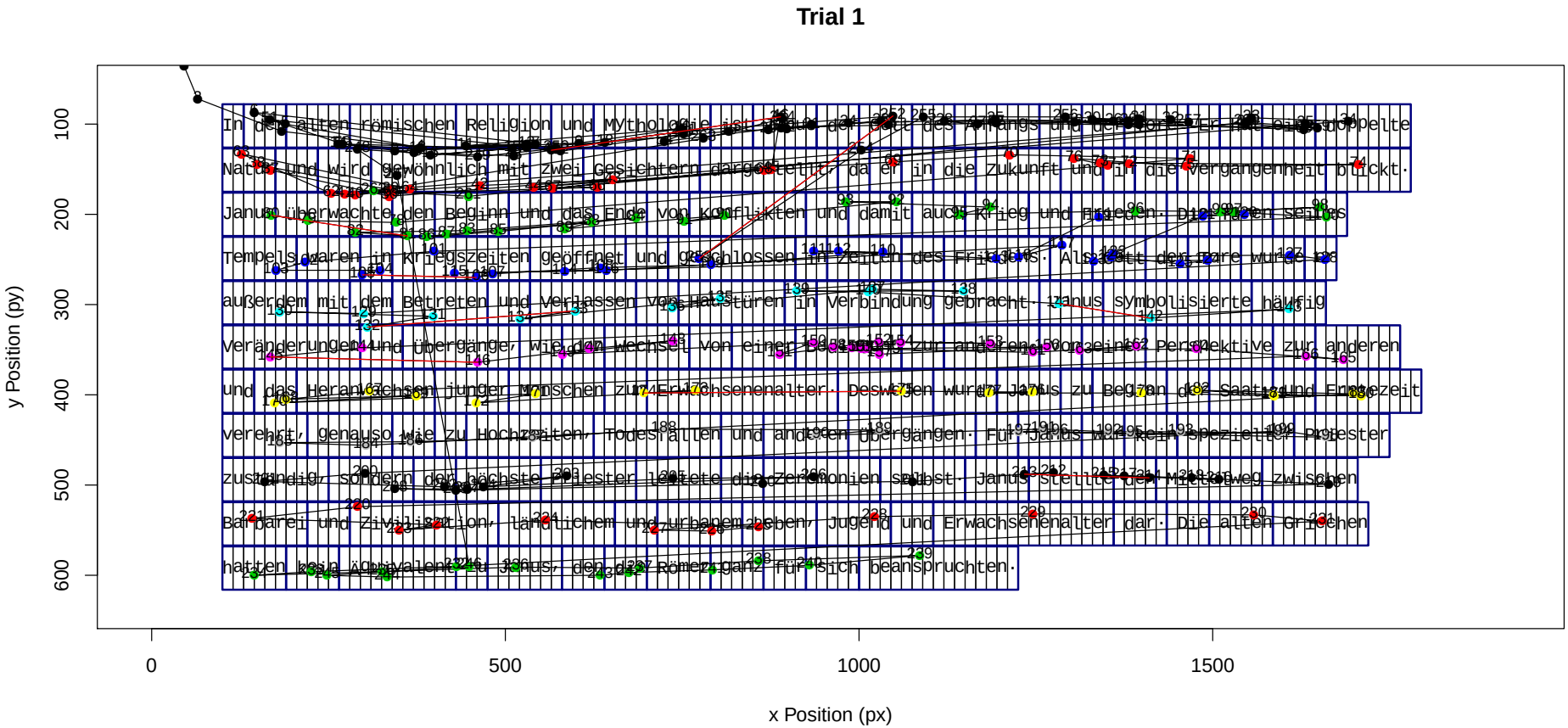
Preprocessing: Assign Stimuli



Preprocessing: Assign Stimuli



Preprocessing: Assign Stimuli



Preprocessing: Assign Stimuli

Different line alignment methods:

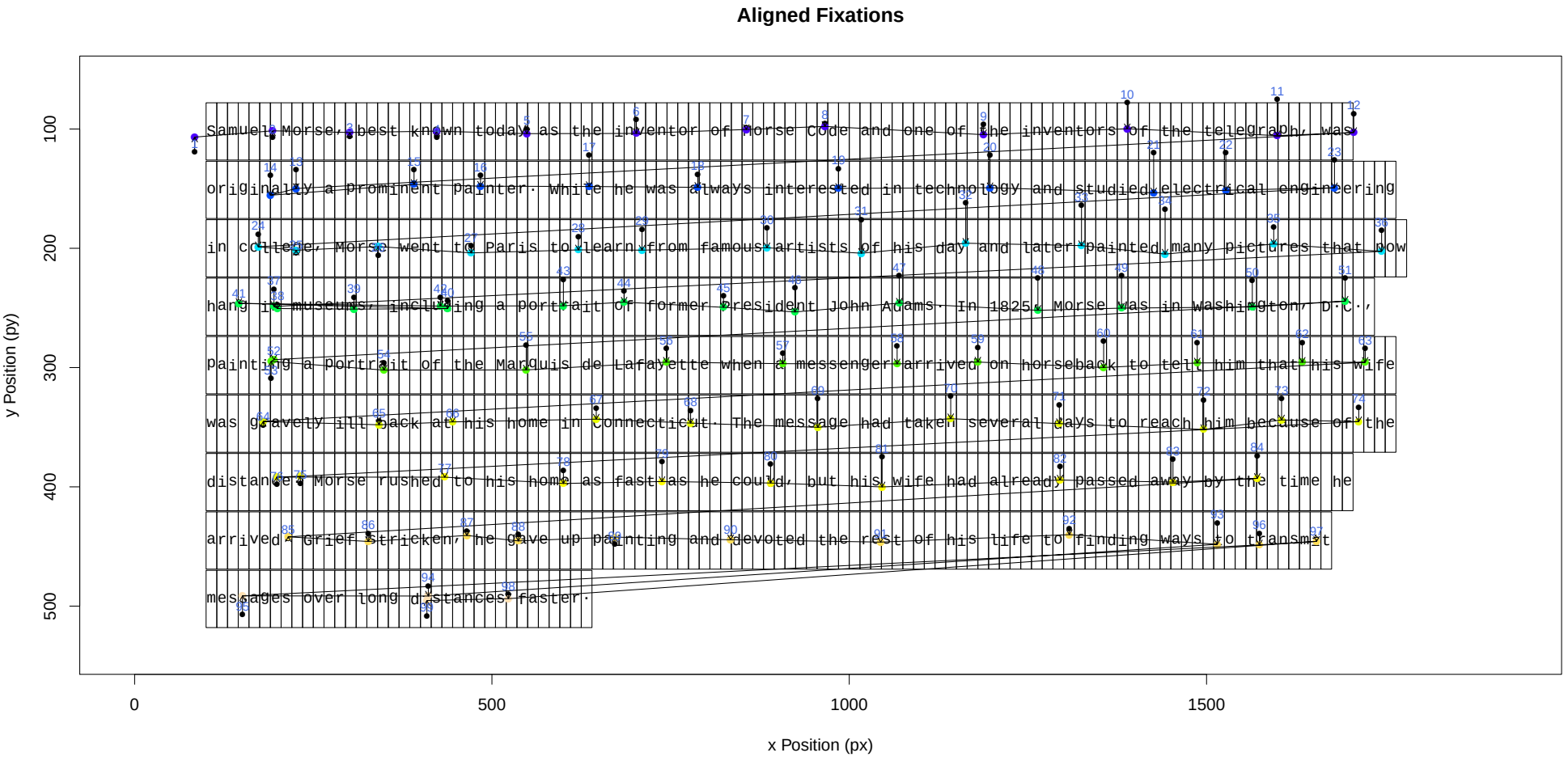
- Distance matching
- Cluster approach
- Chain method

Preprocessing: Assign Stimuli

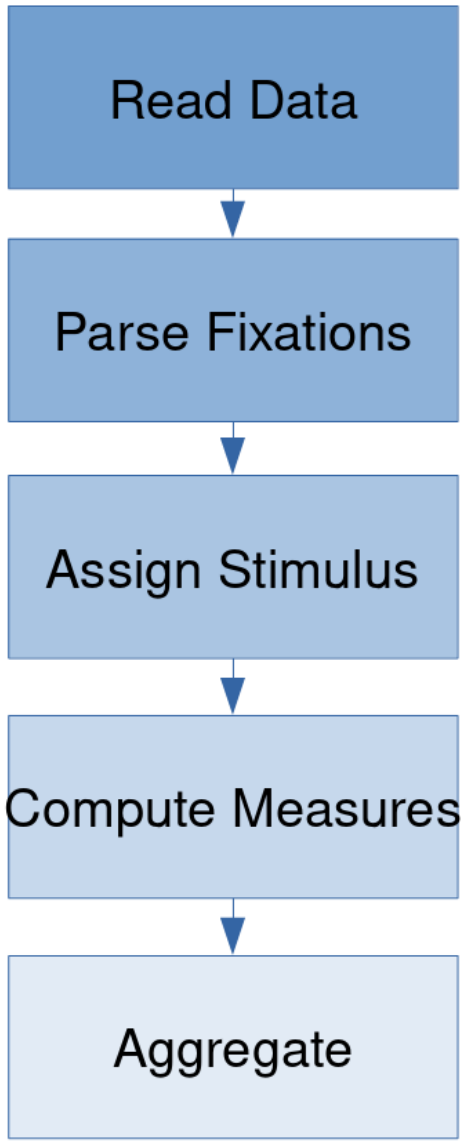
Chain method:

- 1. Segment fixations into runs (sequence of fixations in same direction)
  - Compute distance between fixations on x and y dimension.
  - If x or y distance exceeds a threshold, a new run is initiated.
- 2. Assign runs to lines
  - Mean distance between fixations in run and line position.

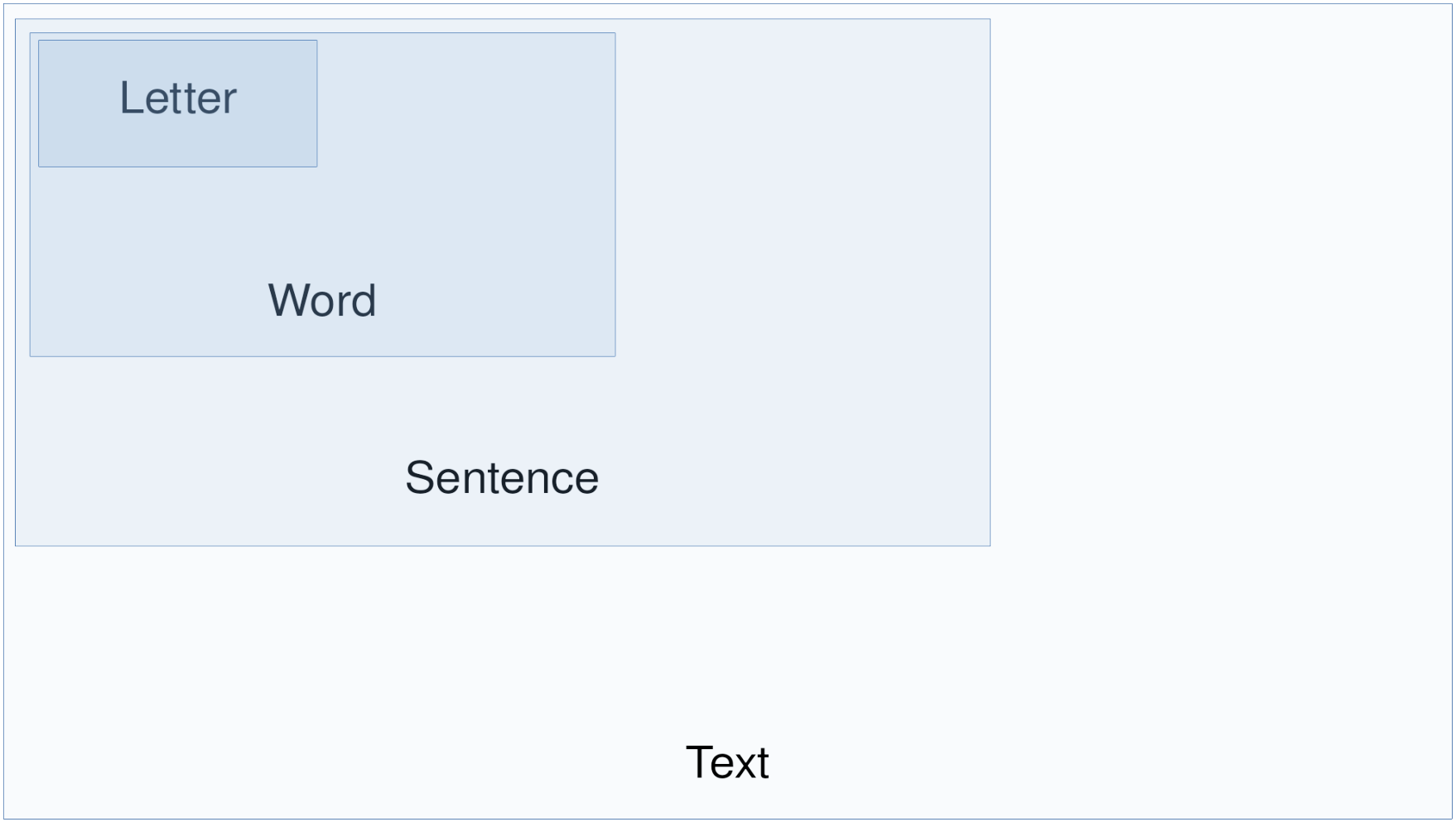
Preprocessing: Assign Stimuli



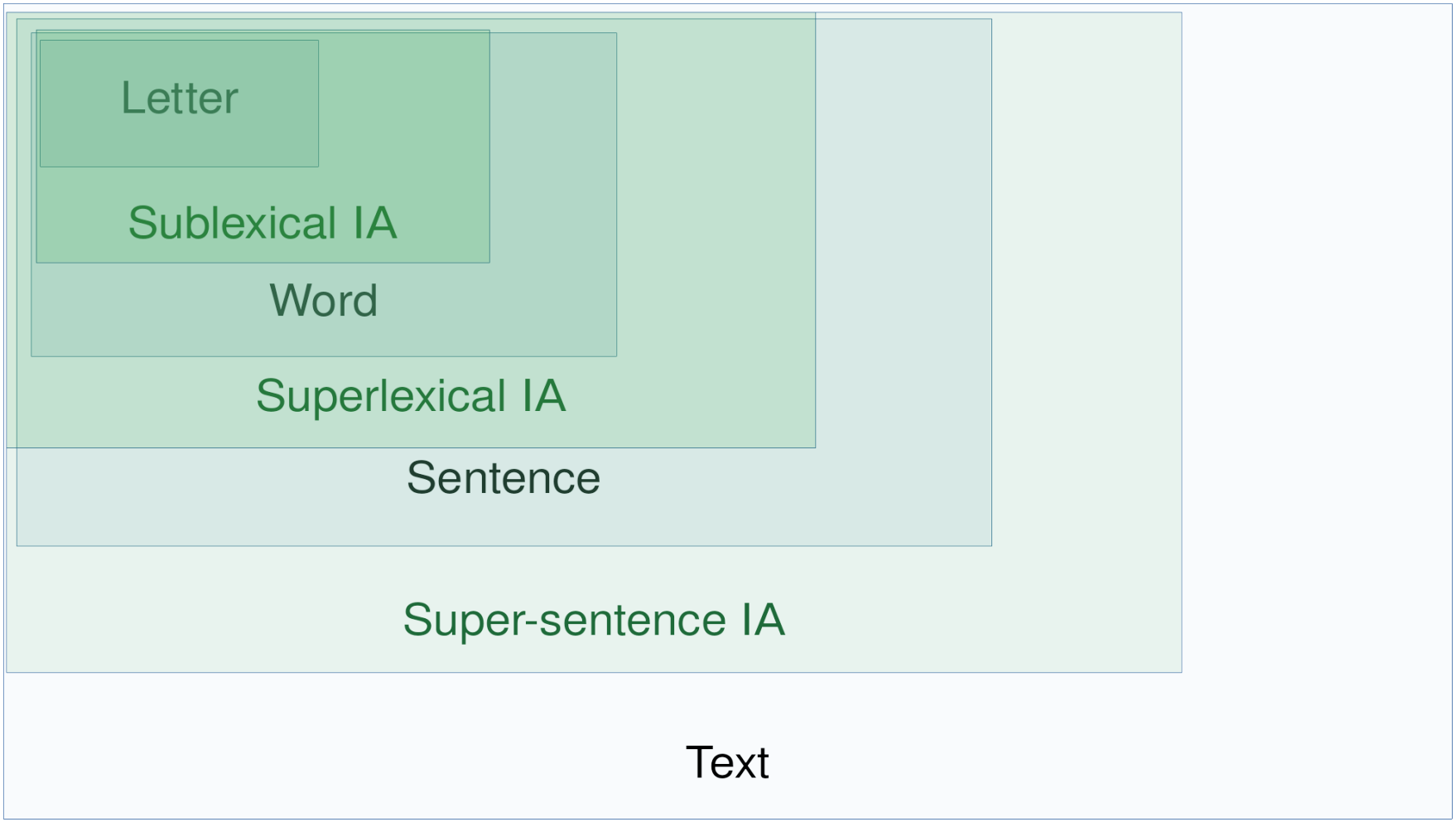
Preprocessing: Aggregate



Preprocessing: Aggregate



Preprocessing: Aggregate



Preprocessing: Aggregate

On each level, fixations are aggregated:

- Firstrun vs. rereading measures
- Skippings, refixations, regressions in/out
- First fixation duration, gopast time, total reading time etc.
- Landing position, launch sites etc.

Computation of level-specific measures: e.g. on sentence- and text-level (Hyönä, Lorch & Rinck, 2003)

Analysis

Preprocessing

- Read Data
- Parse Fixations
- Assign Stimuli
- Compute Measures
- Aggregate

Analysis

- Reports
- Plots
- Cleaning

Analysis

```
library(popEye)

exp <- readRDS("Experiment.RDS")
```

Analysis: Reports

In this file, all raw data for each participant and each trial are still available:

```
str(exp$subjects$subject.der3_01$trials$trial.1)
```

```
List of 8
 $ meta :List of 11
 $ xy   :'data.frame':  5172 obs. of  3 variables:
 $ vxy  :'data.frame':  5172 obs. of  3 variables:
 $ parse:'data.frame':  39 obs. of  9 variables:
 $ fix  :'data.frame':  20 obs. of  66 variables:
 $ sac  :'data.frame':  19 obs. of  17 variables:
 $ all  :'data.frame':  43 obs. of  12 variables:
 $ clean:List of 4
```

Analysis: Reports

Reports are stored as separate data frames:

`exp$out$ia`

	subid	trialid	ianum	ia	target	nrun	nfix	dur	firstrun.skip	firstrun.nfix	firstrun.dur	firstfix.land	firstfix.dur
1	der3_01	1	1	Um	<NA>	1	1	143	0	1	143	0	143
2	der3_01	1	2	die	<NA>	1	2	541	0	2	541	3	162
3	der3_01	1	3	Bodenstruktur	<NA>	2	4	840	0	3	535	2	177
4	der3_01	1	4	aufzulockern	<NA>	1	2	471	0	2	471	1	192
5	der3_01	1	5	wird	n-1	1	1	271	0	1	271	2	271
6	der3_01	1	6	das	n	1	1	259	0	1	259	1	259
7	der3_01	1	7	Feld	n+1	1	1	261	0	1	261	3	261
8	der3_01	1	8	regelmäßig	<NA>	2	3	653	0	2	456	4	291

[. . .]

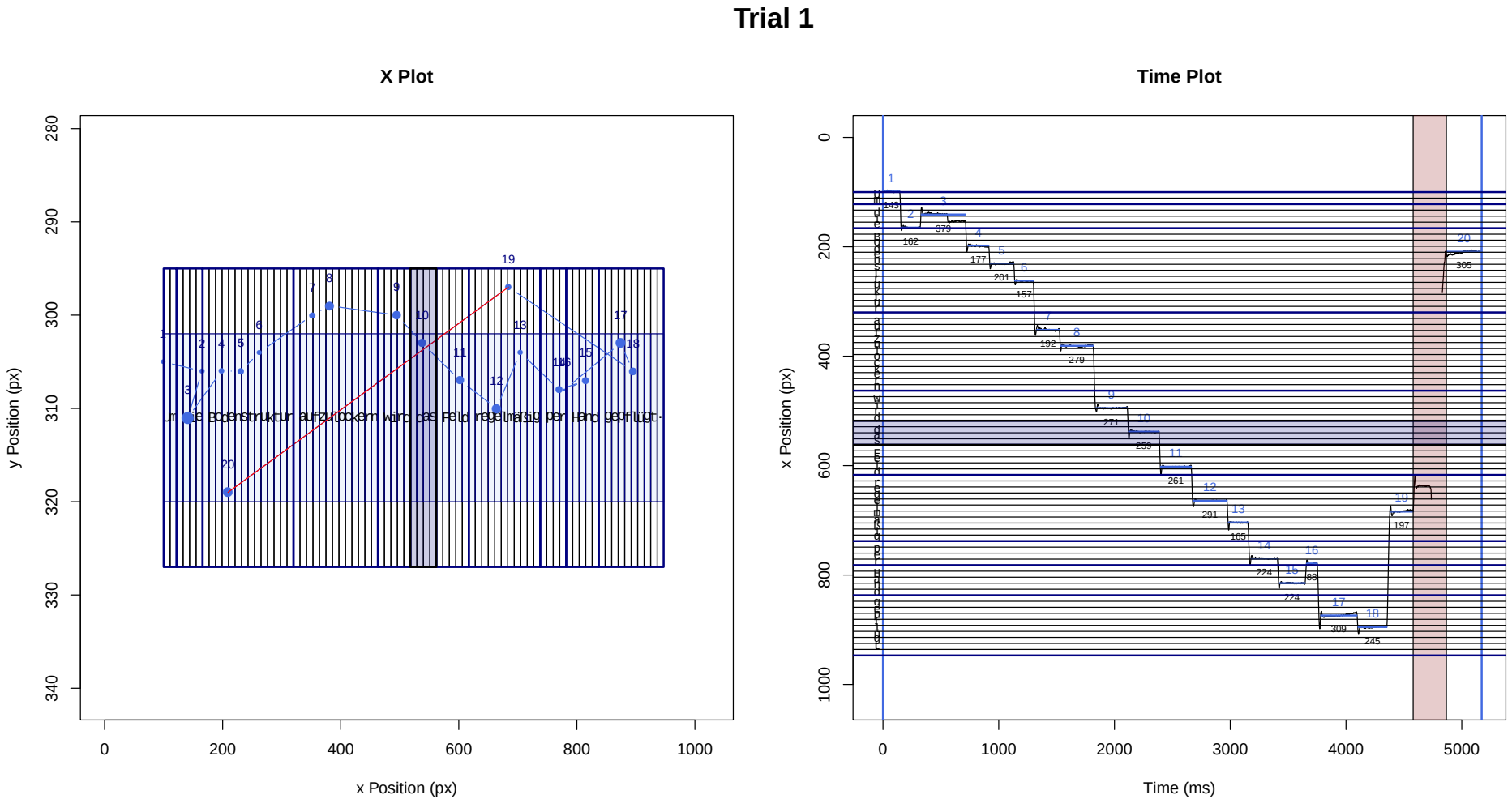
Analysis: Reports

Different report levels:

- Participants
- Text/Trial
- Sentence
- IA
- Word
- Fixation
- Saccade

Analysis: Plots

```
PlotTarget(exp, sub = "sub1", trial = 1)
```



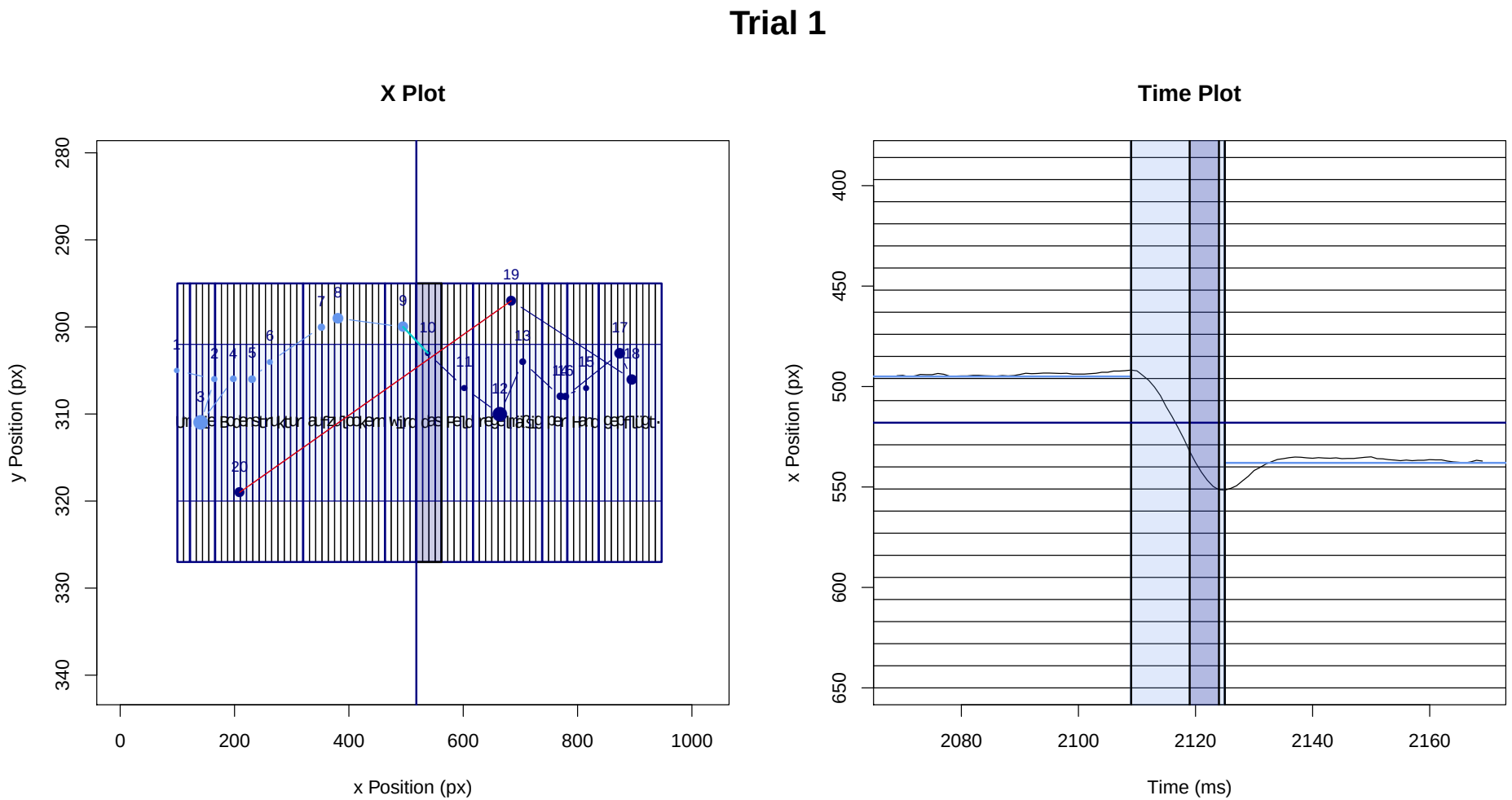
Analysis: Plots

Different plot types:

- Text
- Sentence
- Target
- Boundary
- Fast Priming

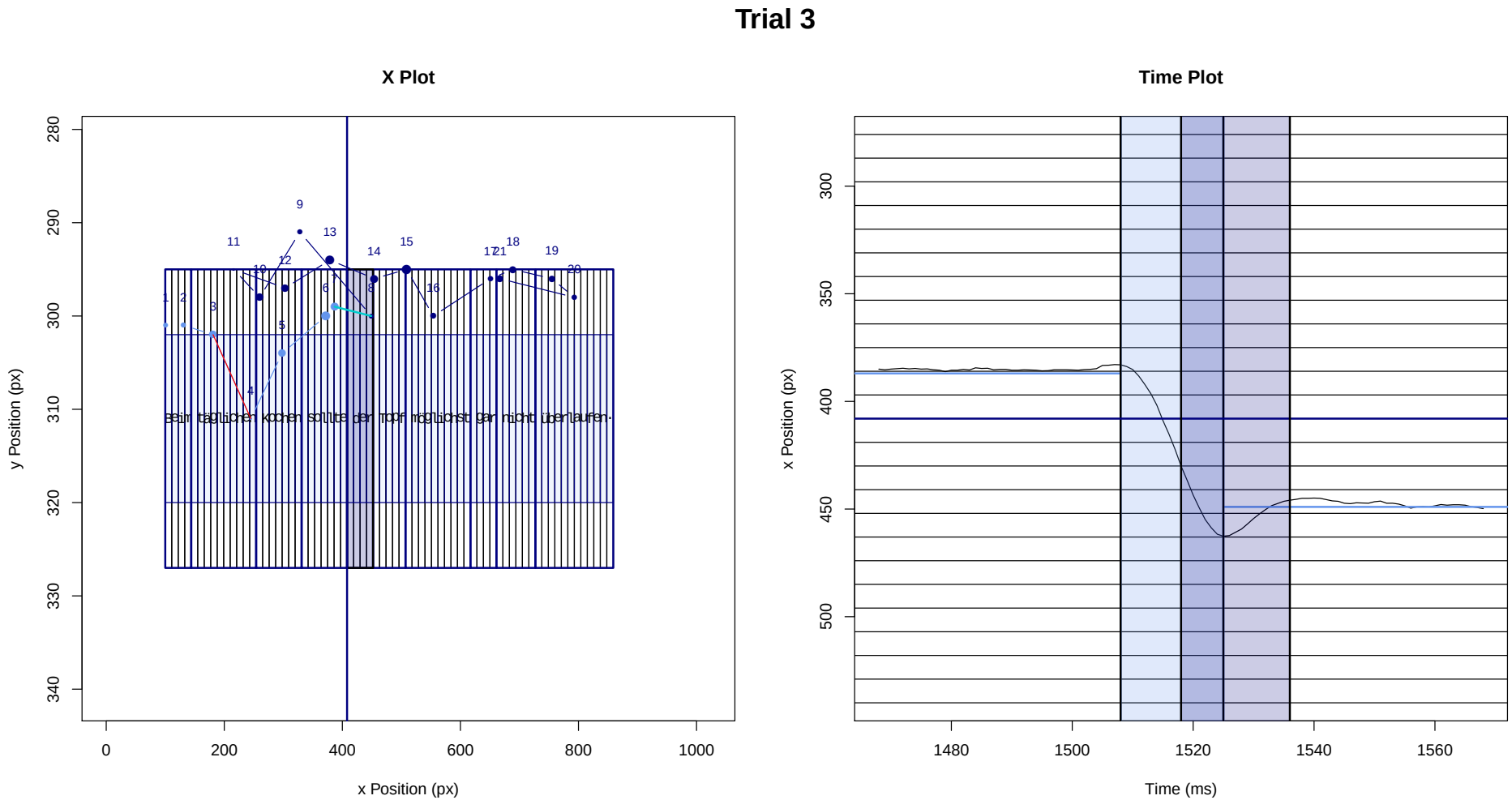
Analysis: Plots

```
PlotBoundary(exp, sub = "sub1", trial = 1)
```



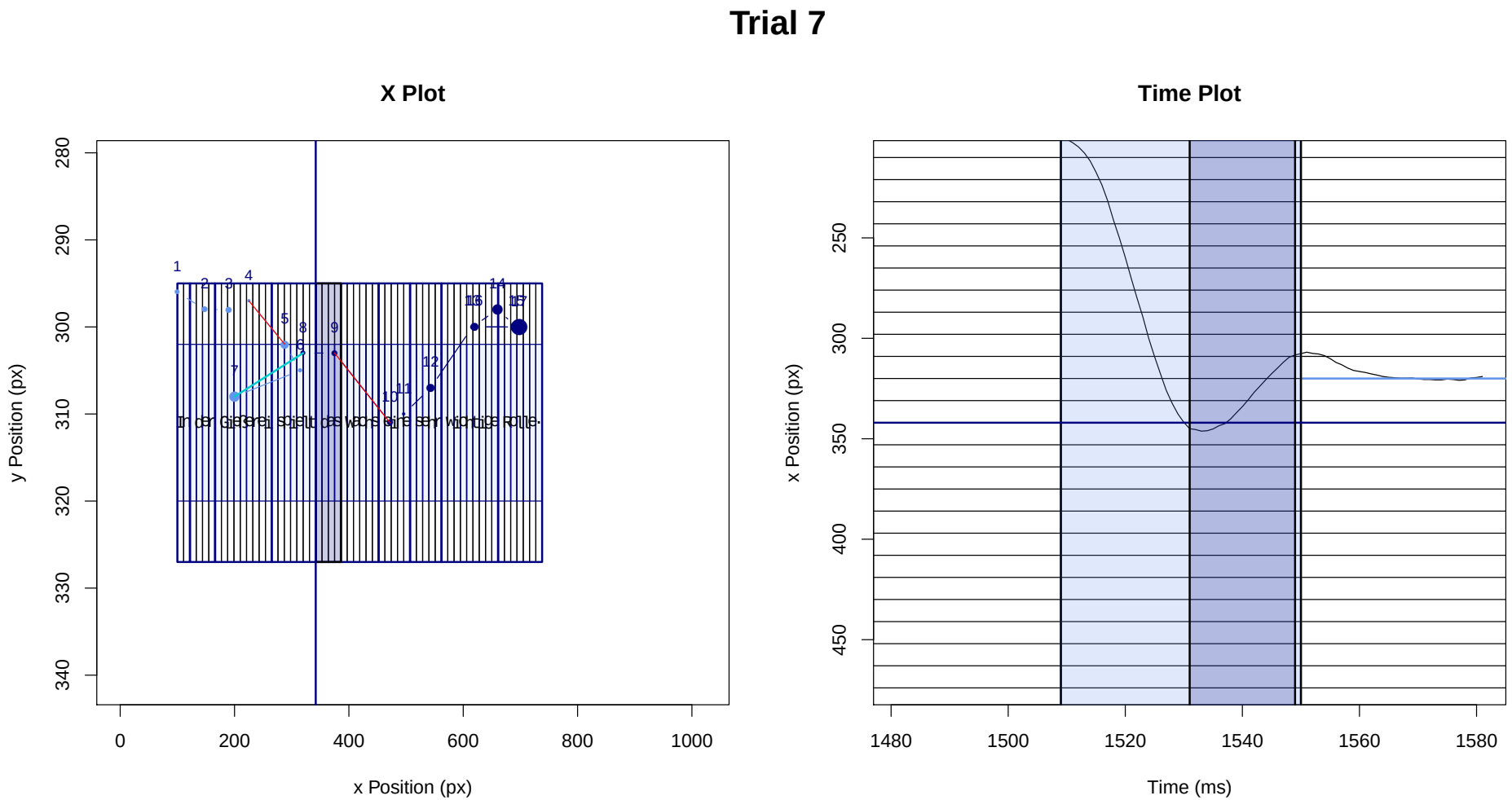
Analysis: Plots

```
PlotBoundary(exp, sub = "sub1", trial = 3)
```



Analysis: Plots

```
PlotBoundary(exp, sub = "sub1", trial = 7)
```



Analysis: Cleaning

```
exp$out$clean
```

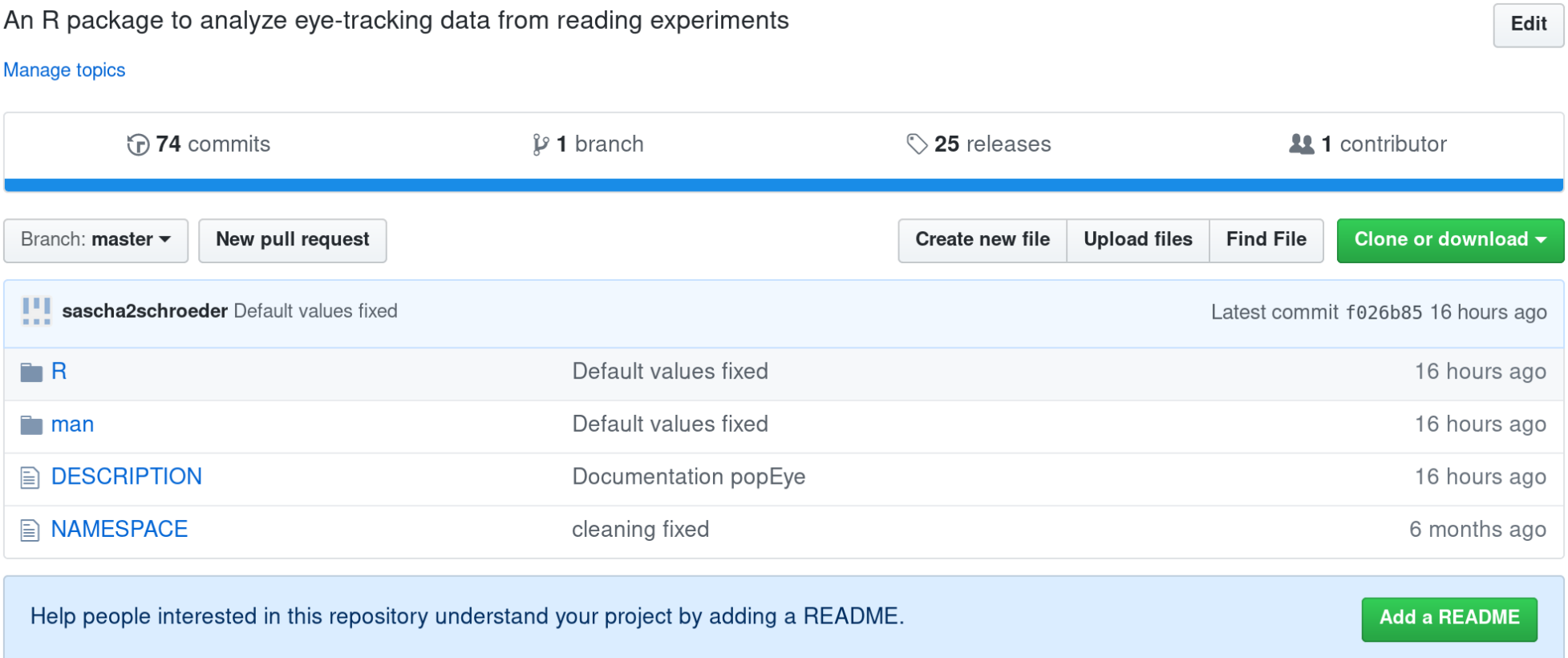
	subid	trialid	boundary.pre.time	boundary.post.time	boundary.time	boundary.hook	boundary.crit
1	der3_01	1	9	1	0	0	0
2	der3_01	2	12	-2	0	0	0
3	der3_01	3	9	-11	1	0	1
4	der3_01	4	14	2	0	0	0
5	der3_01	5	10	-11	1	0	1
6	der3_01	6	NA	18	0	1	1
7	der3_01	7	21	1	0	1	1

[. . .]

(When) Should I care?

- If you are interested in text- and paragraph-reading studies...
- If you are interested in boundary-change paradigms...
- If you want to have more control over your analysis...

Where do I find popEye?

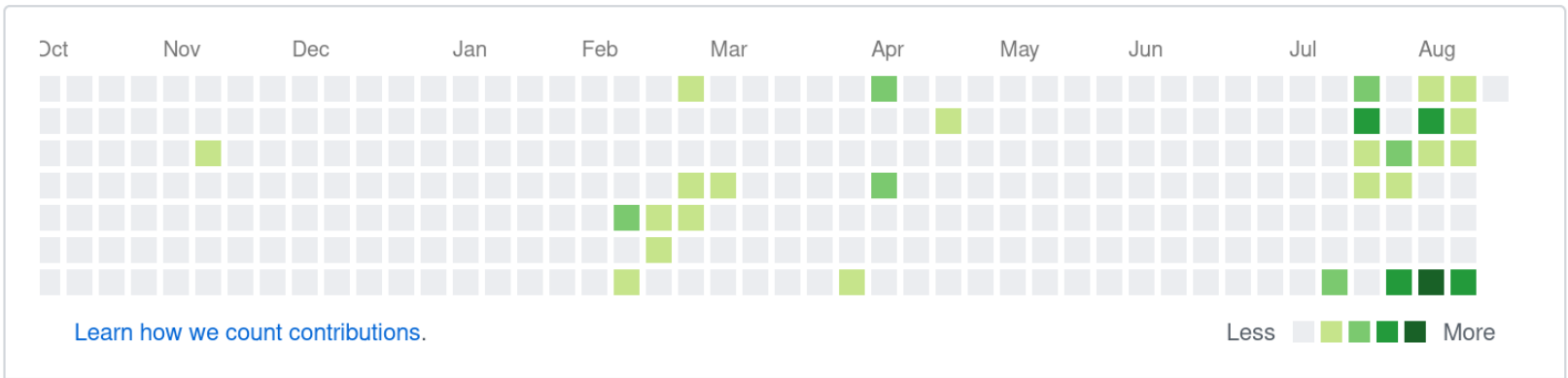


Please contribute, test, give feedback - any help is appreciated!

Next Steps

- Documentation!
- Increase scope (more fonts etc.).
- Meta-level functions.
- Adapt for PyGaze.
- Adapt for SMI and GazePoint.

Timeline



- August 2019 0.6.0: Current version
- September 2019 0.7.0: Documentation
- October 2019 0.8.0: Beta release

Thank you for your attention!

sascha.schroeder@psych.uni-goettingen.de