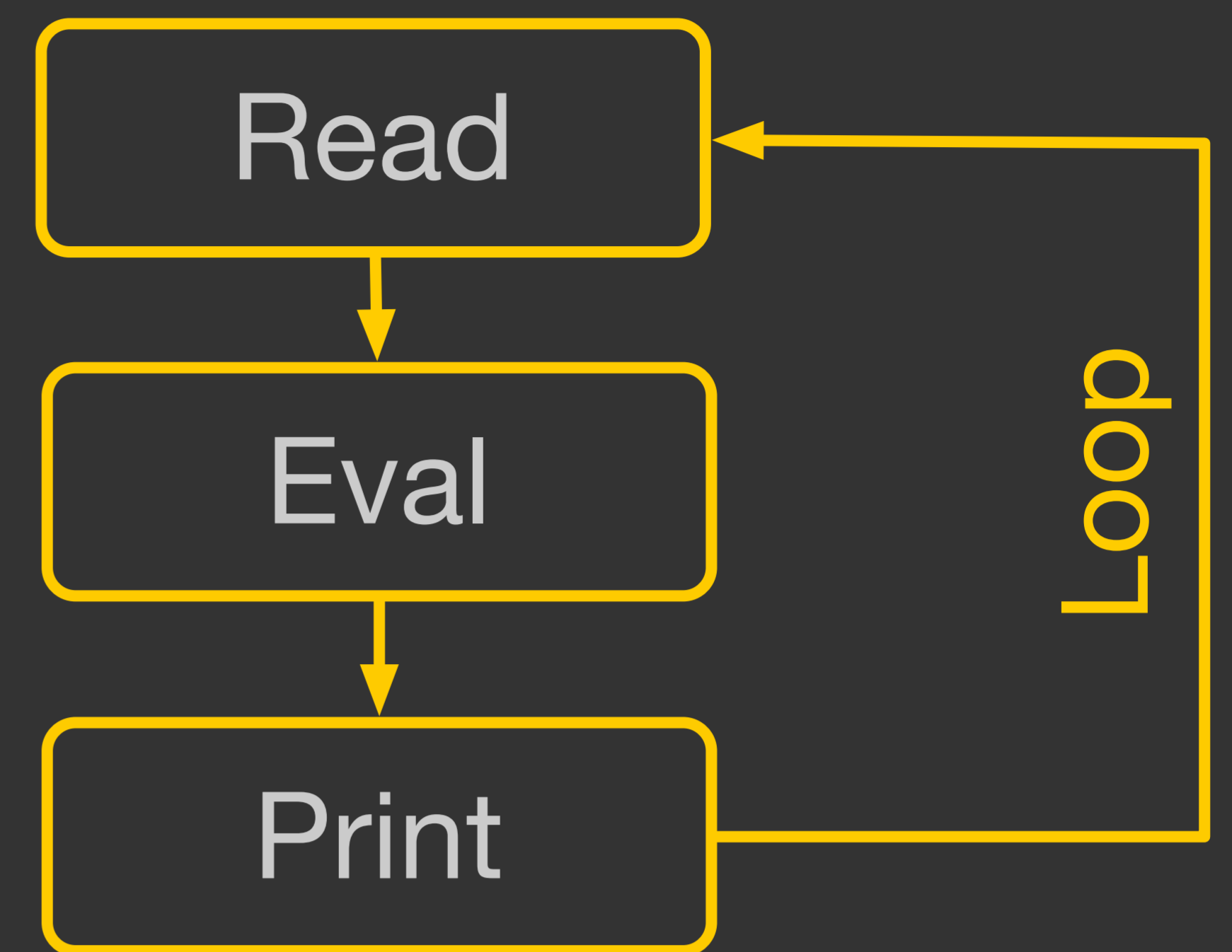


Designing an IDE for Probabilistic Programming: Challenges and a Prototype

Sameer Singh¹, Luke Hewitt², Tim Rocktäschel², Sebastian Riedel²
University of Washington University College London

Need for an PPL-IDE

- Machine Learning cannot be *program once and done*
- User needs to iterate on:
 - data preprocessing, cleaning/pruning, visualizing
 - exploring different models
 - choice of the inference algorithms
 - model and inference (hyper)-parameters
 - bugs, often not easily distinguishable from features in ML



Design Challenges an IDE for PPLs

Structure Data & Marginals

- PPL implies structured data
- need to visualize structured input to ensure it's correct
- support task domains such as networks, text, images, etc.

Strutured Marginals

- predictions are also structures, but in fact they are more
- they are structured marginals!
- how can we visualize those?

Model & Inference

- cater to user expertise
- different levels of feedback

Example of Model Feedback

- beginners: summaries
- intermediate: plate models
- experts: debug structure

Example of Inference Feedback

- beginners: confidence
- intermediate: objective plots
- experts: samples/messages

Language Agnosticism

- supporting multiple PPLs is key
- contrast and compare PPLs
- share debugging interface

Intermediate Representation

- active area of research
- none exists yet

API for PPL Visualization

- inference execution trace
- probability distributions
- graphical version of the model

A Prototype for Wolfe

A browser-based, easy-to-use IDE that supports iterative probabilistic programming

REPL Interface

```

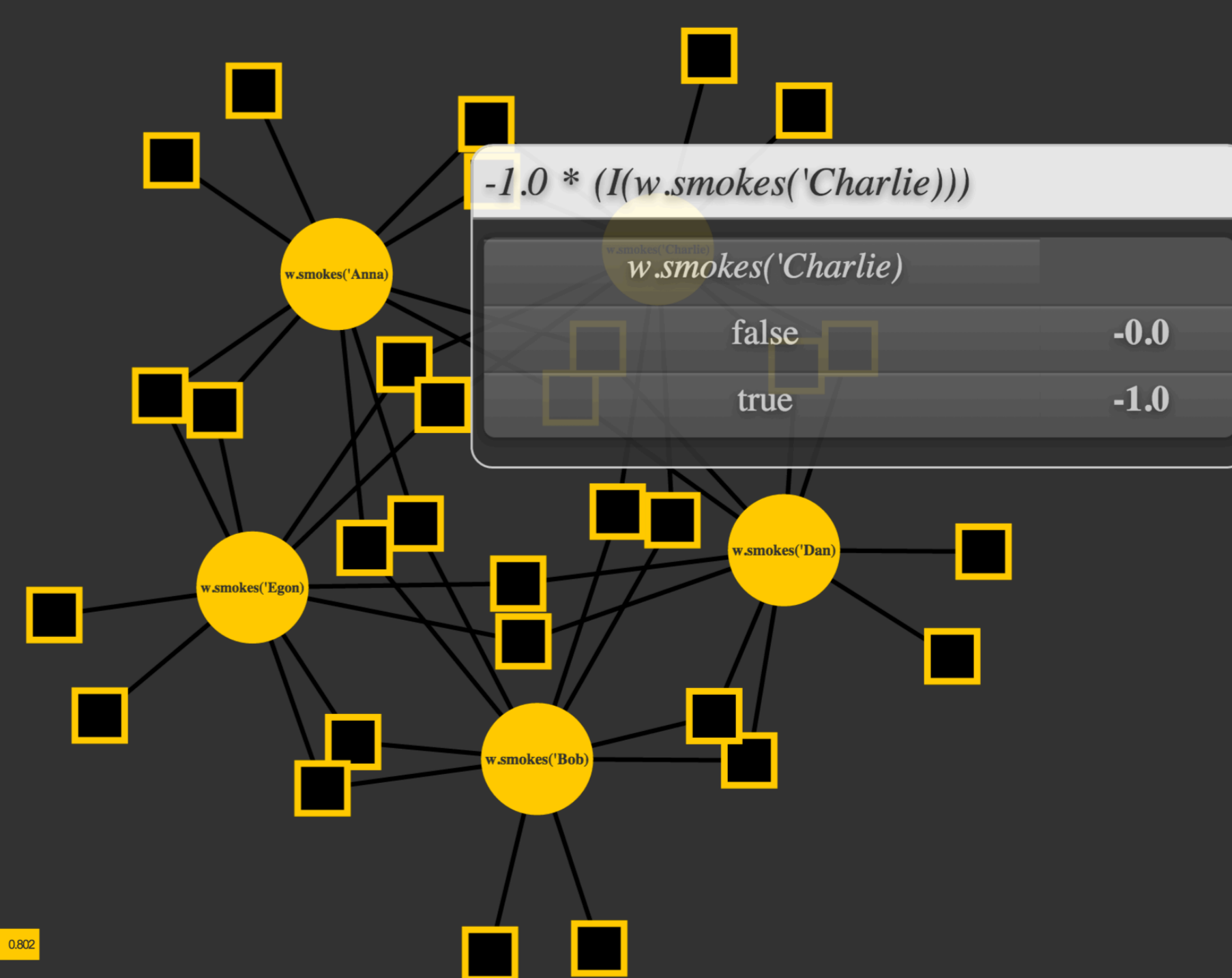
1 val pred = argmax(space(words)) { crf(weights, words) }
2 appendMentions(doc, pred)
    
```



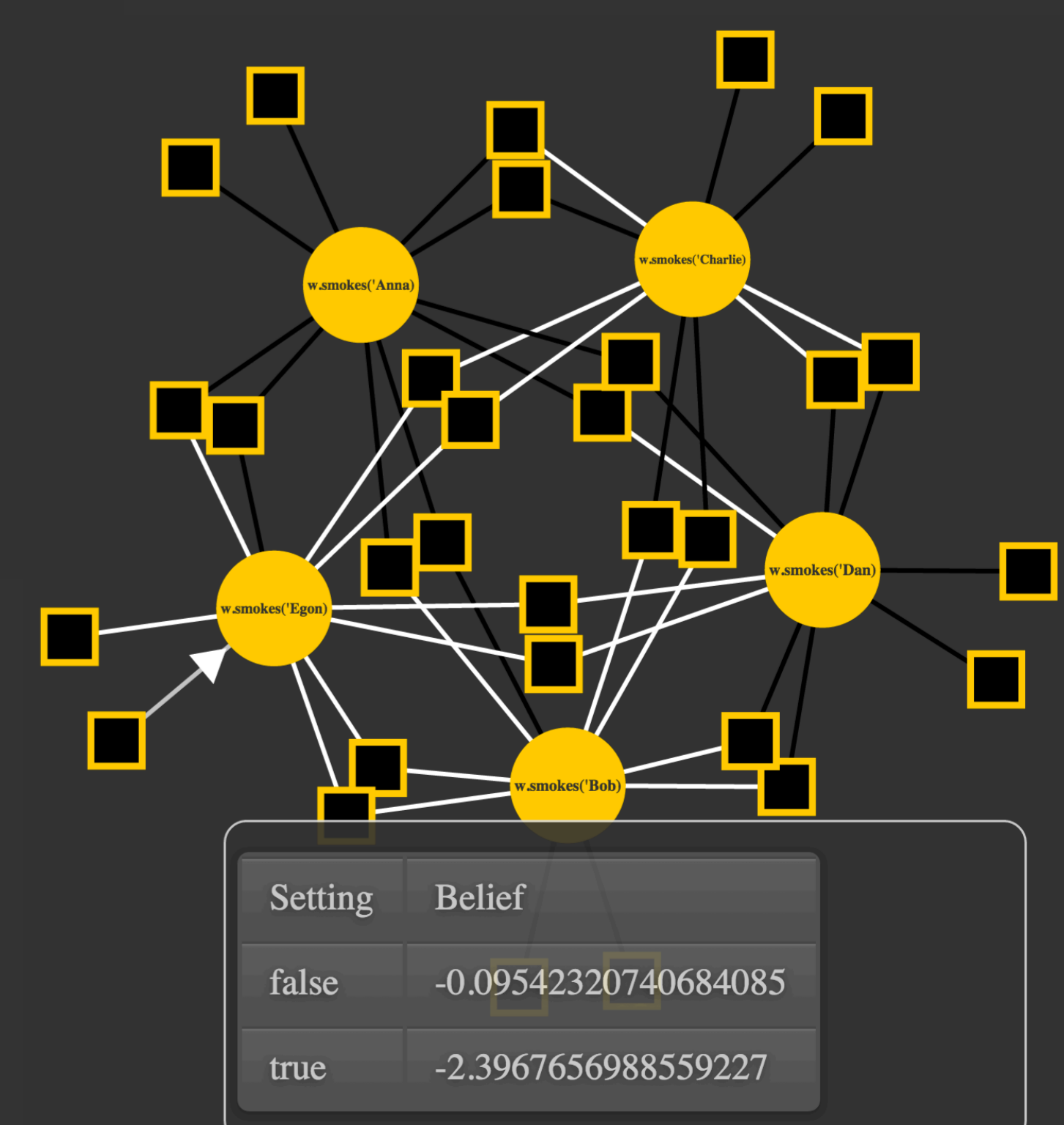
1 **PER**
John Denver is a Songwriter.

2 **LOC**
Throughout his life, Denver produced many records.

Model...

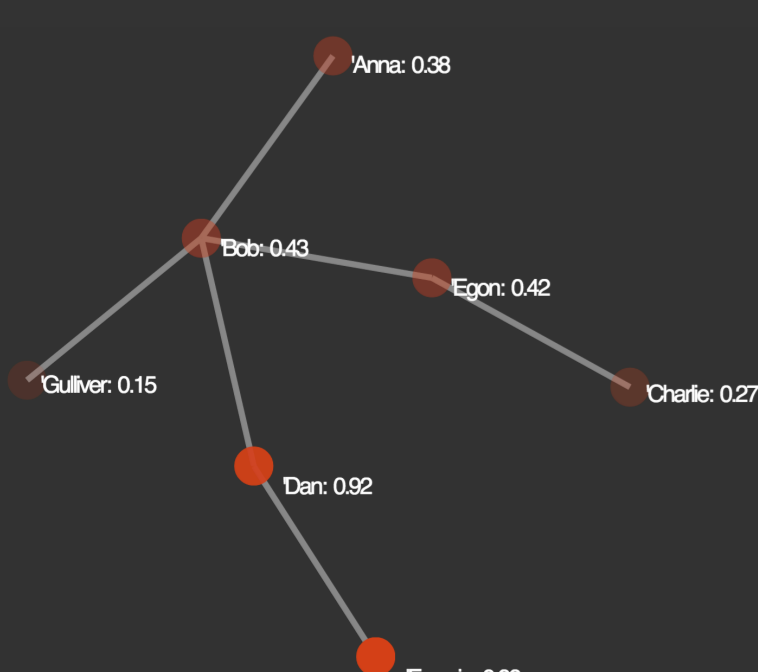


... and Inference

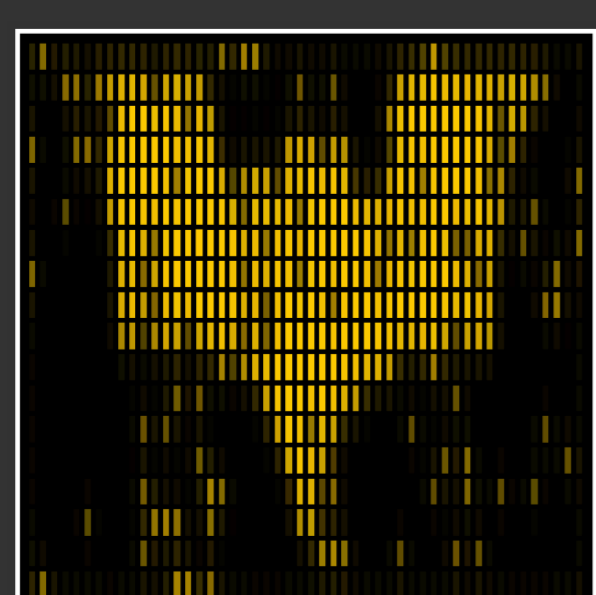


Data and Marginals

Graphs



Matrices



Plots

