BioNet - Functional analysis of biological networks

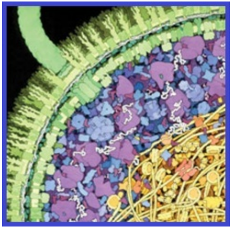
Daniela Beisser, Gunnar W. Klau, Thomas Dandekar,
Tobias Mueller, Marcus Dittrich

Department of Bioinformatics
University of Würzburg

Integrated network biology

- Integration of different data sets becomes more and more important

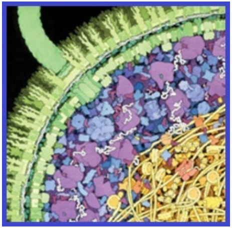
Wordle - ISMB2010 highlight talks



Integrated network biology

- Integration of different data sets becomes more and more important
- Biological networks to put genes or proteins in context

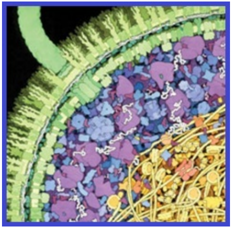
Wordle - ISMB2010 highlight talks



Integrated network biology

- Integration of different data sets becomes more and more important
- Biological networks to put genes or proteins in context

Wordle - ISMB2010 highlight talks



The components of a biological system do not act independently from each other, but are organized into functional modules
(Hartwell et al., 1999)

Integrated network analysis

DNA - Genome

- Promotor Analysis
- Gene Regulation
- CGH Data
- SNPs

Integrated network analysis

DNA - Genome

- Promotor Analysis
- Gene Regulation
- CGH Data
- SNPs

Phenome

- Survival Analysis

Integrated network analysis

DNA - Genome

- Promotor Analysis
- Gene Regulation
- CGH Data
- SNPs

Phenome

- Survival Analysis

RNA - Transcriptome

- Gene Expression
  - Microarrays
  - RNA-Seq
- Non-coding RNA

Integrated network analysis

DNA - Genome

- Promotor Analysis
- Gene Regulation
- CGH Data
- SNPs

Phenome

- Survival Analysis

RNA - Transcriptome

- Gene Expression
  - Microarrays
  - RNA-Seq
- Non-coding RNA

Proteins - Proteome

- Homology
- Function Prediction
- Protein-Protein-Interaction

Integrated network analysis

DNA - Genome

- Promotor Analysis
- Gene Regulation
- CGH Data
- SNPs

Phenome

- Survival Analysis

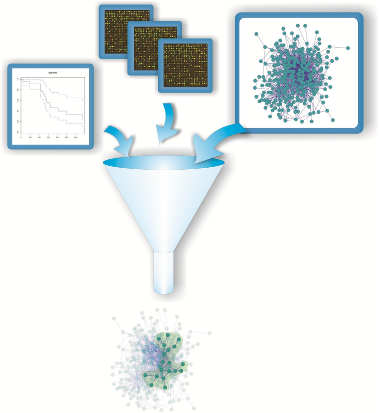
RNA - Transcriptome

- Gene Expression
  - Microarrays
  - RNA-Seq
- Non-coding RNA

Proteins - Proteome

- Homology
- Function Prediction
- Protein-Protein-Interaction

BioNet



Integrate into protein-protein interaction network (HPRD):

- Gene expression data
- Survival data
- ...

To find coherent significant modules.

Workflow

5 steps

Workflow

5 steps

1. Statistical tests

Workflow

5 steps

1. Statistical tests
2. P-value aggregation

Workflow

5 steps

1. Statistical tests
2. P-value aggregation
3. Fit Beta-Uniform-Mixture (BUM) model

Workflow

5 steps

1. Statistical tests
2. P-value aggregation
3. Fit Beta-Uniform-Mixture (BUM) model
4. Score nodes of the network

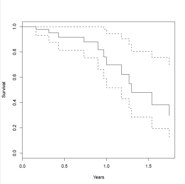
Workflow

5 steps

1. Statistical tests
2. P-value aggregation
3. Fit Beta-Uniform-Mixture (BUM) model
4. Score nodes of the network
5. Find maximum scoring subnetwork

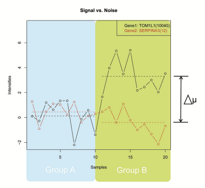
1. Statistical tests

Survival Analysis



- Cox regression

Microarray Analysis



- Standard t-test

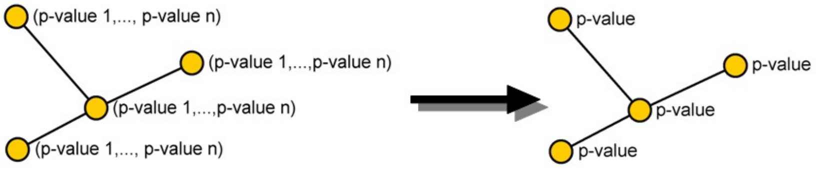
```
for(i in 1:dim(expdat[pat,])[2]{
cox <- summary(coxph(as.formula(paste("sv ∼ ‘",
colnames(expdat[pat,])[i] ,"‘", sep="")),
data=expdat[pat,][i]))

survival.pvalues[i] <- cox$logtest[3]}
```

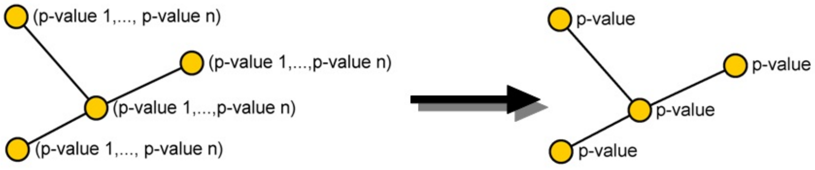
```
ttest.pvalues <- rowttests(exprLym, fac =
exprLym$Subgroup)$p.values
```

For each test: p-value per gene

2. P-value aggregation


(p-value 1,..., p-value n)
(p-value 1,...,p-value n)
(p-value 1,...,p-value n)
(p-value 1,..., p-value n)
p-value
p-value
p-value
p-value

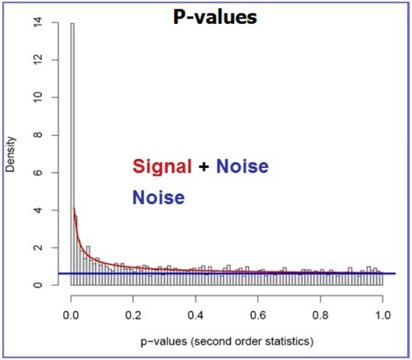
2. P-value aggregation


(p-value 1,..., p-value n)
(p-value 1,...,p-value n)
(p-value 1,...,p-value n)
(p-value 1,..., p-value n)
p-value
p-value
p-value
p-value

- P-values are uniformly distributed
- i^{th} Uniform Order statistic of n p-values is Beta distributed
- Regard X(1) → at least one significant p-value
- Regard X(n) → all p-values significant

```
pvals <- cbind(ttest.pvalues, survival.pvalues)
pvalues <- aggrPvals(pvals, order=2, plot=F)
```

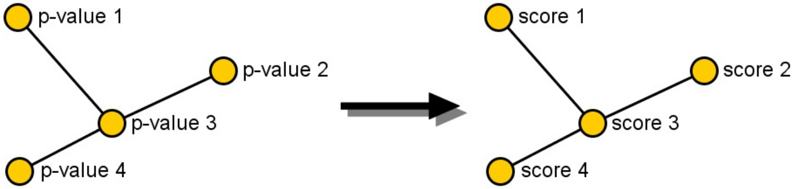
3. Fit Beta-Uniform-Mixture (BUM) model



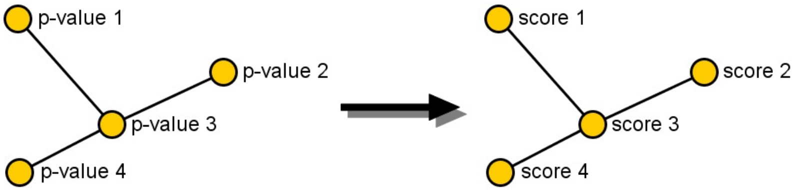
$$f(x) = \lambda \; Noise(x) + (1 - \lambda) \; Signal(x; a)$$

```
fb <- fitBumModel(pvalues, plot=TRUE)
```

4. Score nodes of the network



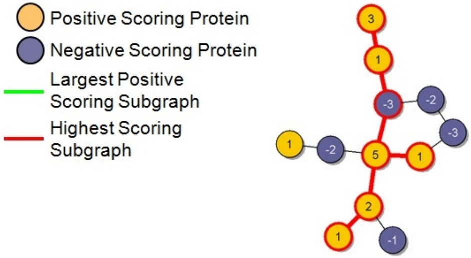
4. Score nodes of the network



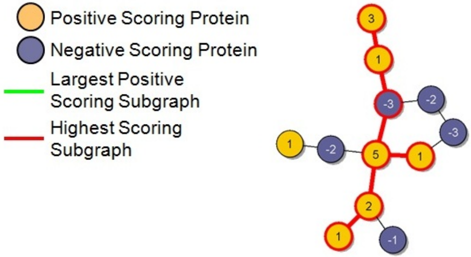
$$S(x) = log\frac{Signal(x)}{Noise(x)} = log\frac{Beta(a,1)(x)}{unif(0,1)(x)} = log(a) + (a-1)log(x)$$

```
scores <- scoreNodes(network=net, fb=fb, fdr=0.001)
```

5. Find maximum scoring subnetwork


● Positive Scoring Protein
● Negative Scoring Protein
Largest Positive Scoring Subgraph
Highest Scoring Subgraph
3
1
-3
-2
-3
1
-2
5
1
2
1
-1

5. Find maximum scoring subnetwork



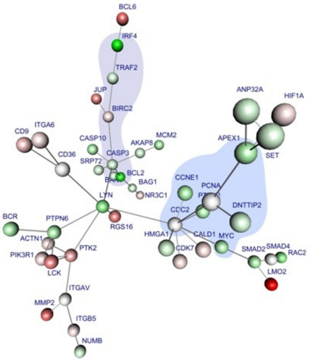
- NP-hard → ILP
- Solution by combinatorial optimization using CPLEX library
- Detection of suboptimal solutions
- Heuristic approach

```
writeHeinzFiles(network=net, file="ex", node.scores=scores)
module <- readHeinzGraph(node.file="ex_n.txt.0.hnz", network=net)
module.heur <- runFastHeinz(network=net, scores=scores)
```

Maximal-scoring subnetwork for DLBCL

red nodes
↑ upregulated
in GCB

green nodes
↑ upregulated
in ABC



```
plot3DModule(module)
```

Acknowledgements