

Machine learning applied to the prediction of transcriptional regulation in prokaryotes

- predicting control logic, operons (and genenetworks) -

Danny Arends

S1276891

Dept of Molecular Genetics

Supervisor: Prof. O. Kuipers

Direct Supervisor: Dr. S. van Hijum



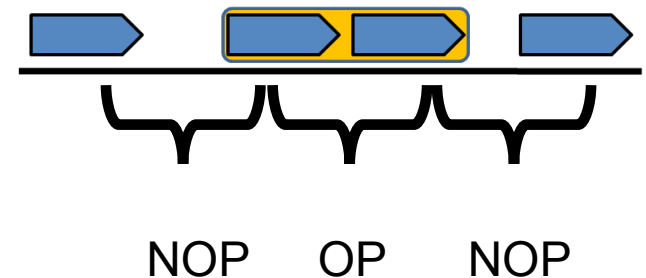
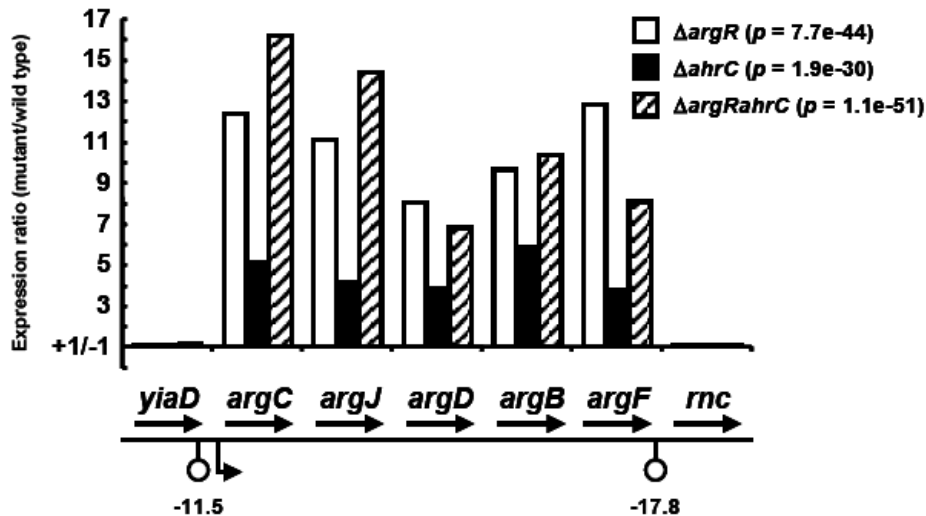
Topics

- Introduction
 - Operons
 - Regulons, control logic
 - [Network reconstruction]
 - Classification
- Methods
- Results
- randomForest web-tool
- Conclusion
- Discussion
- Future perspectives

Operon prediction

- Operons are the smallest transcriptional units.
- Functional relationship between genes in operon
- Operon gene-pair (OP)
 - Intergenic distance, same direction, co-expression

A



Operon prediction associated problems

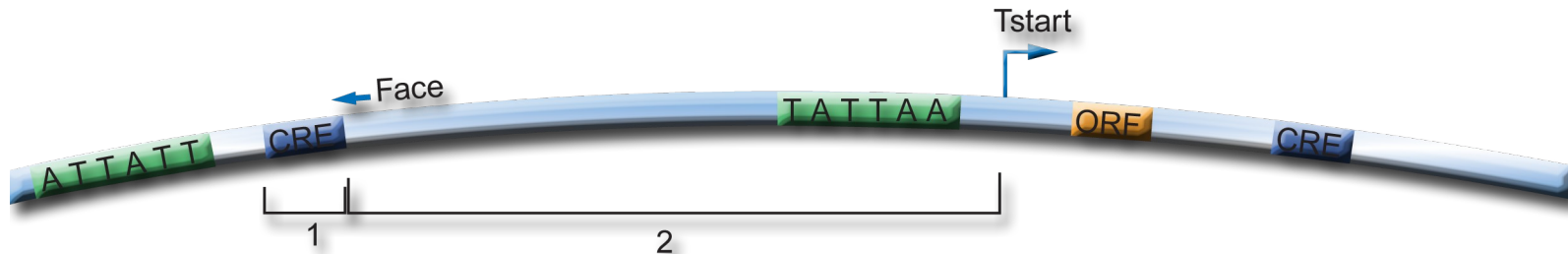
- Different prediction methods
- Low overlap between predictions
- Predictions based on multiple data sources
- Consistency verified operons

Regulon control logic

- Regulon determination
 - Microarray targets
 - Co-expression
 - Consensus sequence
 - Search with consensus for ‘new targets’
 - Lots of false positives (500 in our case)

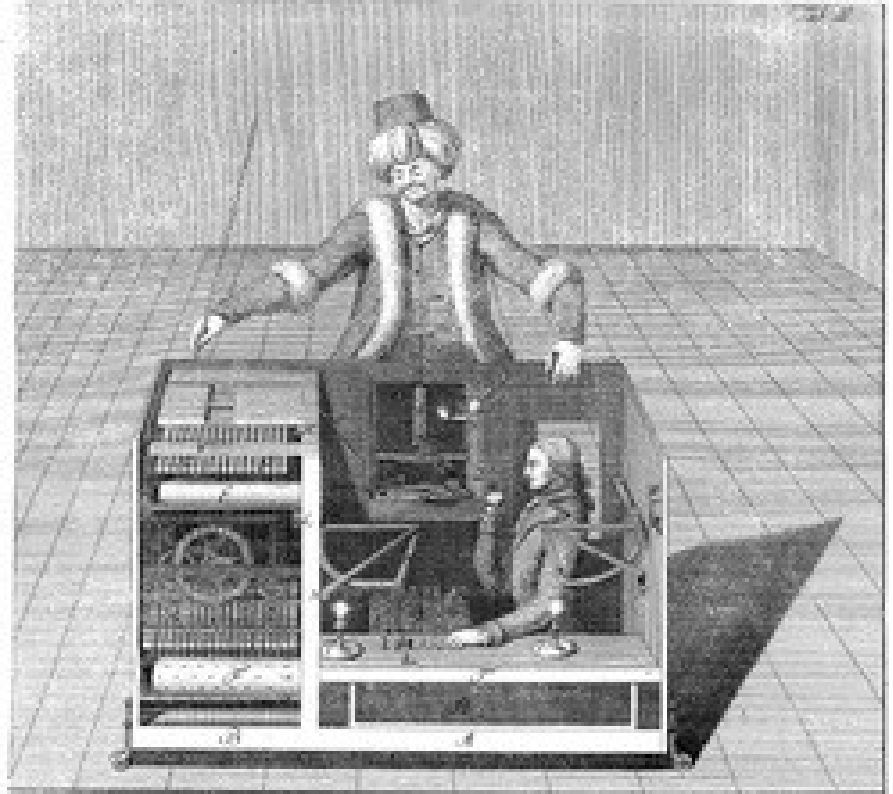
Control logic of CcpA

- In *B. subtilis* about 40 genes are regulated by the transcription factor (TF) CcpA
- The CcpA protein binds to a CRE box
- Orthologous transcription factor in *L. lactis*
- 500 hits CRE consensus search
- Motifs self-fulfilling prophecy



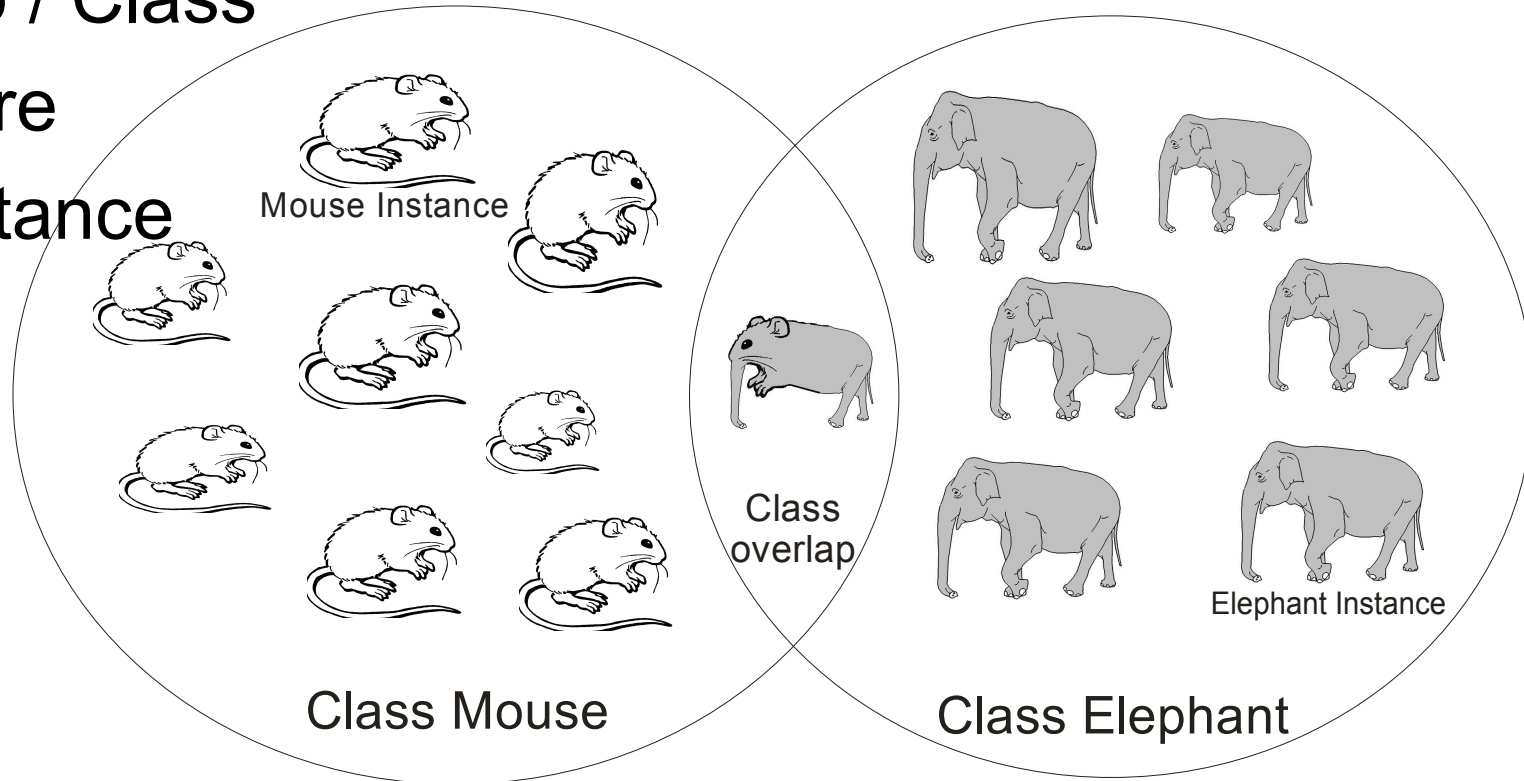
Machinelearning

- Artificial Intelligence
- Mimic human learning
- Internal model
- Learning by example
- Able to make errors



Definitions

- Instance
- Group / Class
- Feature
- Importance



Hypothesis

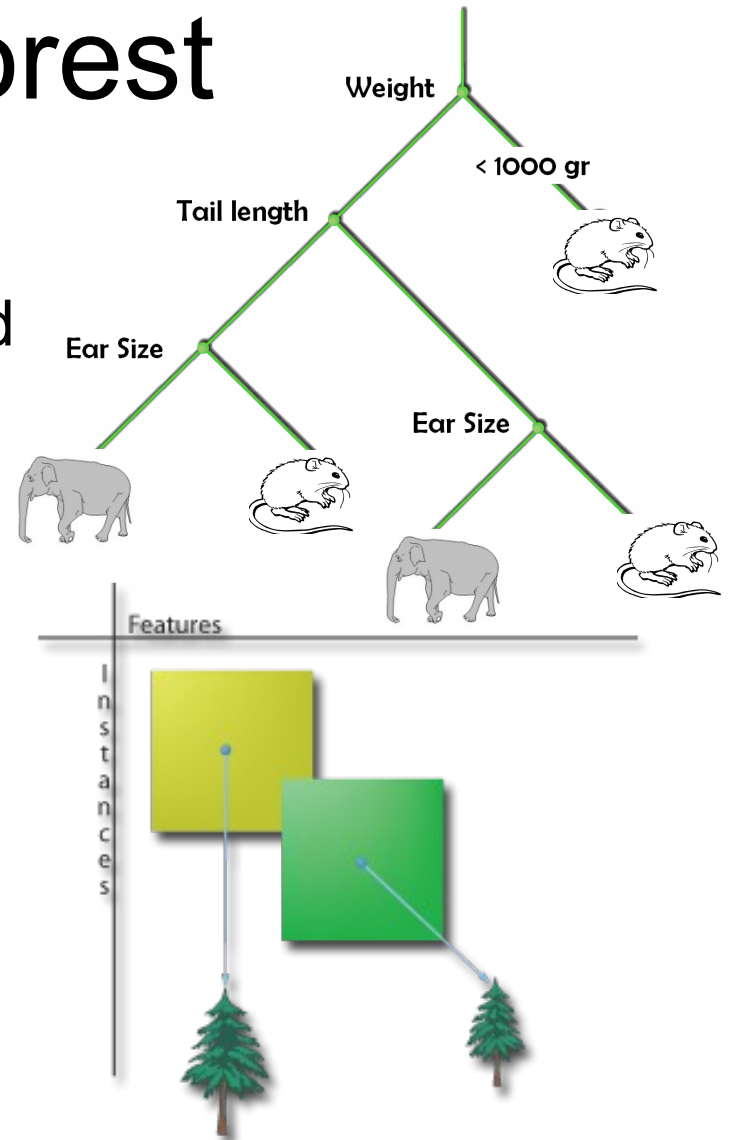
- Machinelearning techniques can help:
 - Provide a generalized method
 - Integrate various data sources
 - Predict which genepairs are in an operon
 - Assign probability of genepairs being an operon
 - Cross species operon prediction
 - Identify which CRE boxes are functional
 - Reduce the number of 'false' positives
 - Probability of box being functionally active

Classification

- Assigning classes to instances based on their features
- Training and Testing
 - Learning by example
 - Classification of new instances
- randomForest
 - Breiman and Cutler
 - Quick, generalized and understandable
 - For small (50) to huge (10.000) number of instances and features
 - Multiclass
- Learning Vector Quantization (not shown)

randomForest

- A forest of decision trees
 - Final class is majority vote based
 - Suitable for multiclass problems
5. Select random 63 % of instances
 6. Select random subset of features
 7. Order features based on class separation
 8. Build a decision tree
- Start with the best separating feature
 - Build next tree from 1.
- All trees together form a Random Forest



randomForest classification

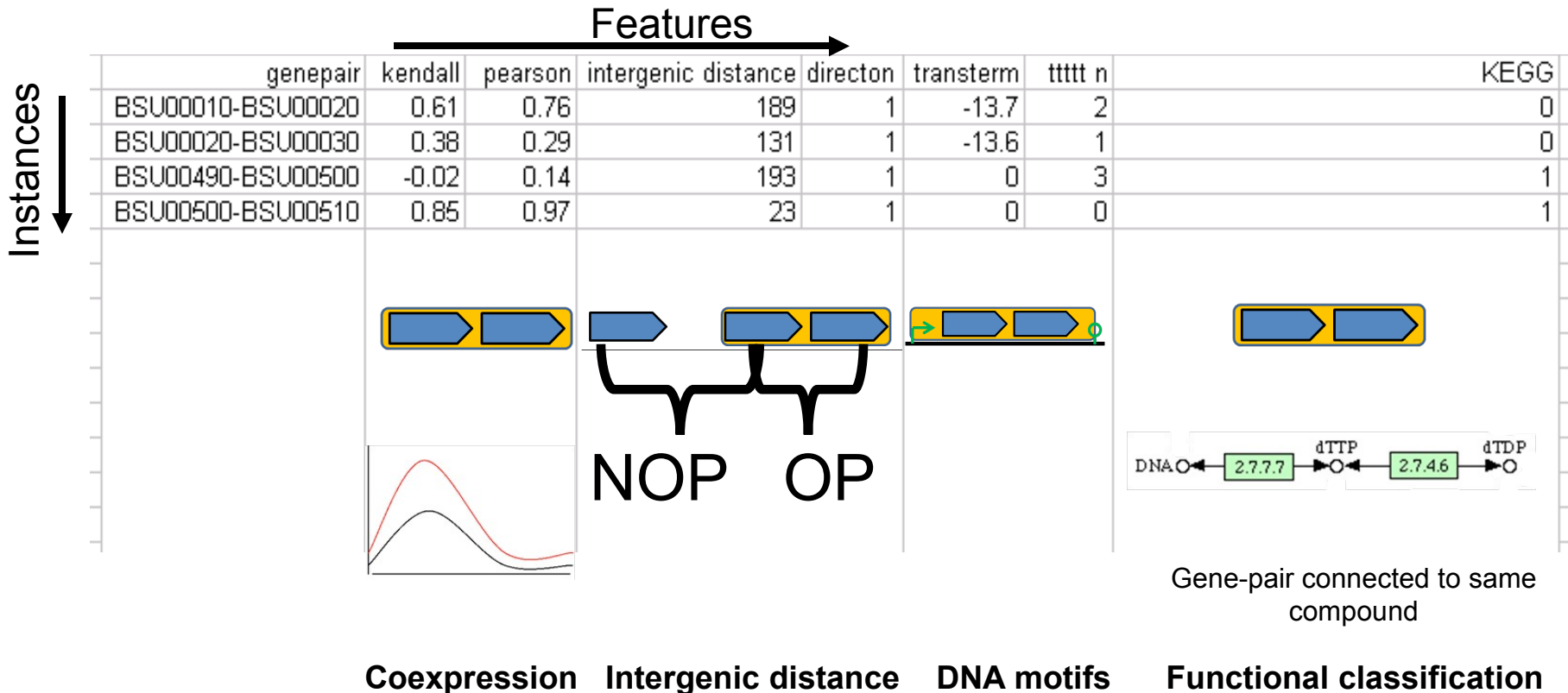
- Each tree votes for a class of an instance
- Majority vote determines final class
- Probability of classification depends on percentage of trees voting for that class
- More tree votes: higher confidence

Materials & Method

- Computer
 - Programming environment (Perl)
 - Microarray data handling
 - Basic calculations / modifications
 - Mathematical environment (R)
 - Package randomForest
- Timeseries microarray data
 - *B. subtilis*, *L. lactis*
- Genetic information
 - *Genome sequences*
- Literature information
 - Verified operons for *B. subtilis*, *L. lactis* & *M. tuberculosis*
 - *B. subtilis* CcpA interactions from DBTBS

Method - Operon prediction

- Create a featurematrix of verified operon pairs

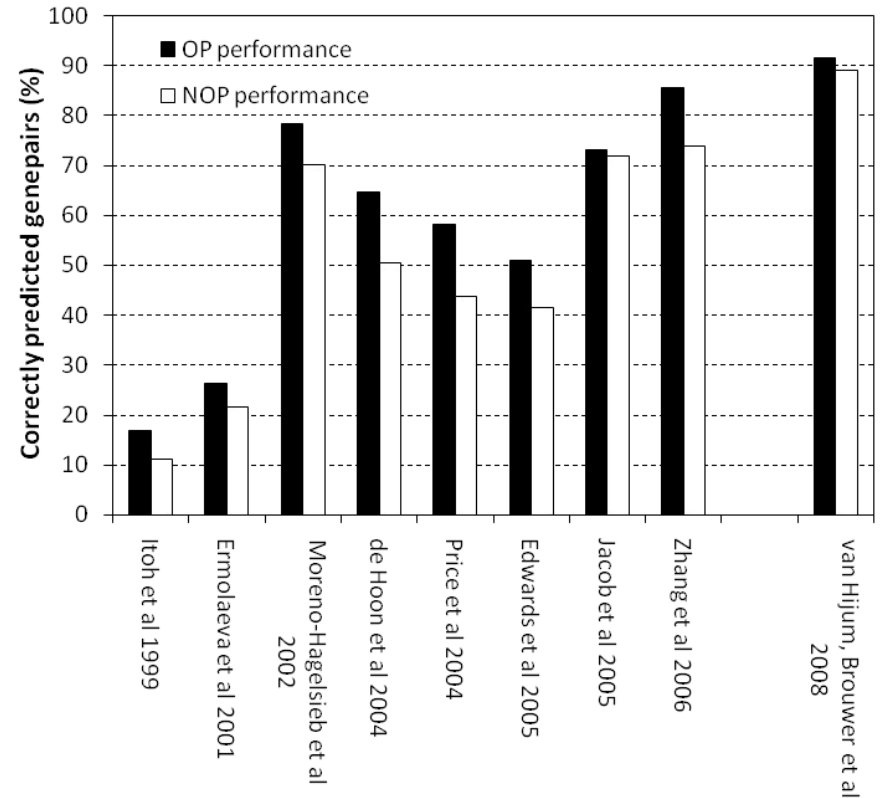
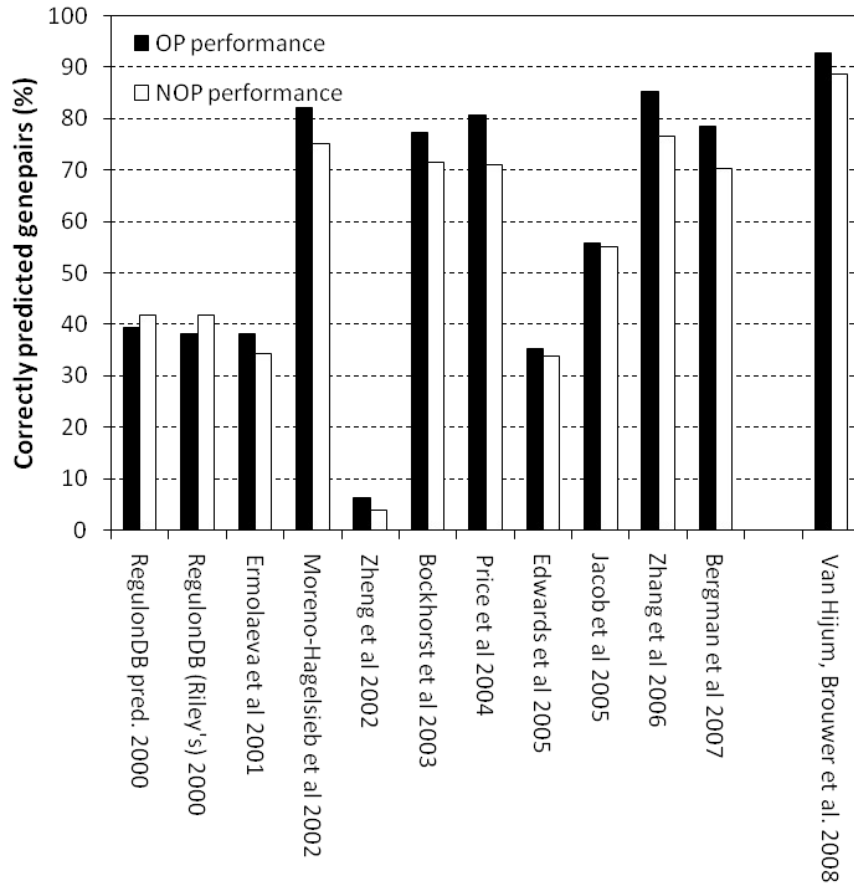


- Train a forest
- Determine performance of classification

Sheet from presentation S. v. Hijum

Results:

Operon prediction compared



Brouwer & van Hijum

Method – Cross validation

Cross species validation

? How generalized is our model ?

- Build featurematrices for:
 - *E. coli* and *B. subtilis* operons
 - Train a forest on *E. coli* known operons
 - Predict (known) operons in *B. subtilis*
 - *Also visa versa*
- Small testset for *M. tuberculosis*
 - Train a forest on combined *E. coli* and *B. subtilis* operons
 - Predict (known) operons in *M. tuberculosis*

Results: cross validation

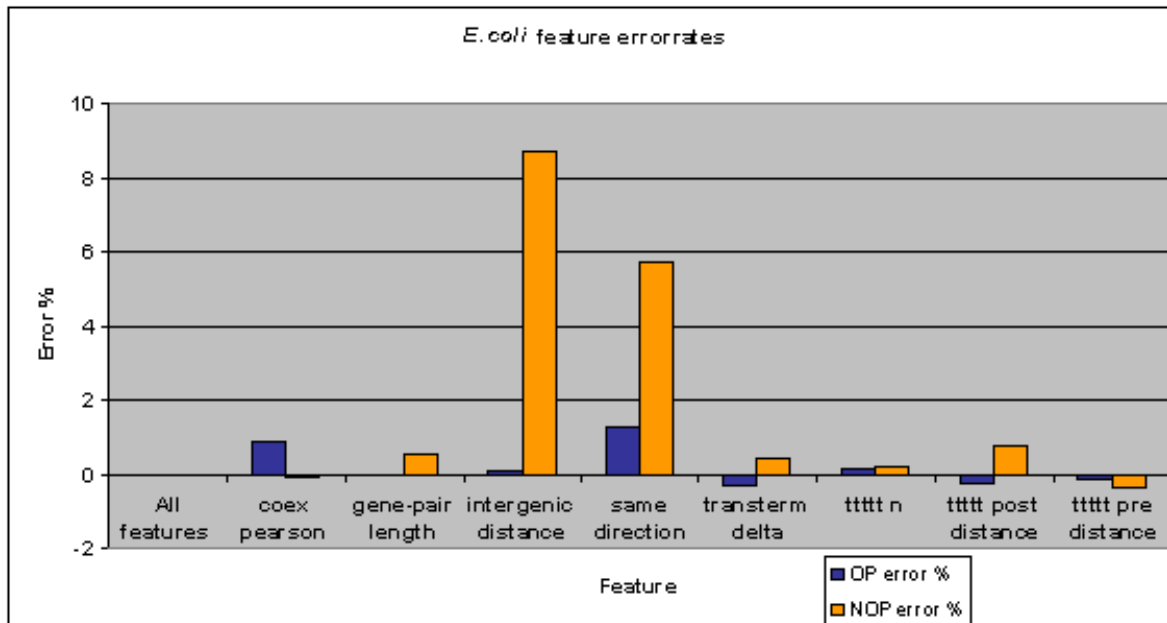
- A RF model is not species dependant -

- *B. subtilis* 656 instances
- *E. coli* 2564 instances
- *M. tuberculosis* 107 instances

Training	Testing	OP error%	NOP error %
<i>B. subtilis</i>	<i>B. subtilis</i>	9.7	11.9
<i>E. coli</i>	<i>E. coli</i>	7.6	11.7
<i>M. tuberculosis</i>	<i>M. tuberculosis</i>	9.4	38.9
<i>B. subtilis</i>	<i>E. coli</i>	17.0	8.0
<i>E. coli</i>	<i>B. subtilis</i>	10.0	16.0
<i>E. coli</i> & <i>B. subtilis</i>	<i>M. tuberculosis</i>	30.0	8.0

Results : feature importance operon prediction

- Estimated during training
- Classification: also available for each instance
- Gain insight into the biological features important for classification.

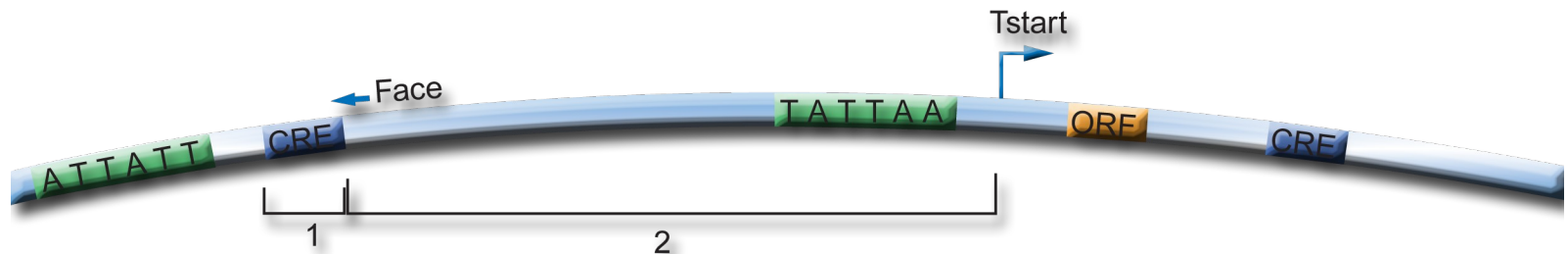


Method - Control logic CcpA

- Determine “true” CcpA members
 - In DBTBS (literature regulon information)
 - Coexpression in DNA microarray experiments
 - Coexpression in timeseries
- Determine highly unlikely CcpA members
 - Not in DBTBS
 - No coexpression
 - Strong CRE motif
- Create feature matrix
- Train a forest
- Determine performance of training

Features for control logic

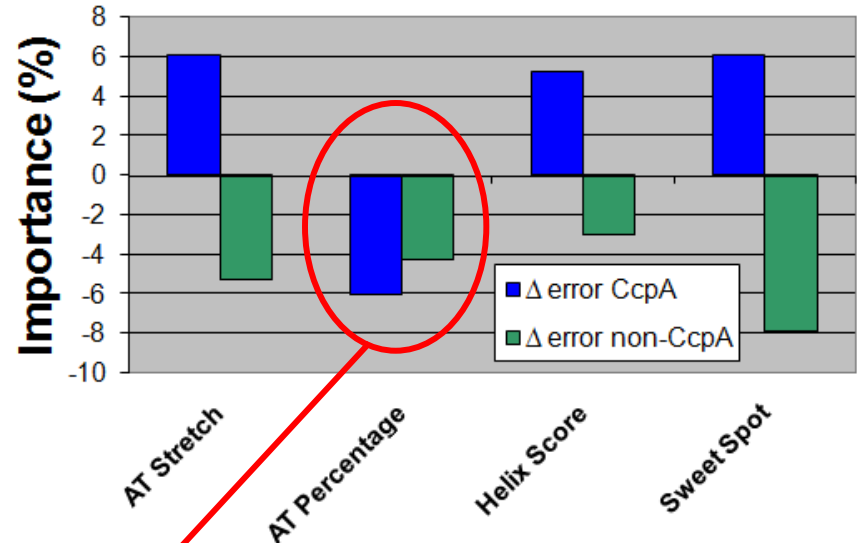
- Many features have been tried
 - BOX features
 - Length CRE box
 - Basepairs in CRE box
 - Palindromicity of CRE box
 - Surrounding features
 - AT-stretch ?
 - Palindrome sequences ?
 - Palindrome stretch



Results: Control logic CcpA *B. subtilis*

- ~15% classification error
- “True” CcpA members
 - AT rich motif
 - Helix face
 - AT stretch (nearby)
- Non-CcpA members
 - No clear separating feature

Classwise feature importance



Confusing feature

	Feature	
	CcpA error (%)	non-CcpA error (%)
With AT perc.	21.2	21.1
Without AT perc.	15.2	16.8

Web Implementation

Web example operon prediction:

- <http://Danny.webdez.nl/FeatureMatrixECOLI.txt>
- <http://bioinformatics.biol.rug.nl/websoftware/rf/>
- http://bioinformatics.biol.rug.nl/websoftware/rf/rf_run.php?session=rf
- Manuals made:
 - RF manual
 - Cytoscape visualization manual

Conclusions

- randomForest
 - Integration of heterogenic data in one model
 - Uncover missing biology (misclassifications)
- CcpA control logic
 - Marked improvement compared to motif
 - No positive selection of inactive CRE
- Operon prediction
 - Strong improvement compared to literature
 - Generalized model: detect operons in different organisms

Discussion & future perspectives

- Application of randomForest to:
 - Gene network reconstruction (not shown)
 - DNA:
 - Control logic – more features?, other TF
 - RNA:
 - Structure determination / enzymatic activity
 - Proteins:
 - Localization – membrane protein or cytoplasmatic protein
 - Structure – do these aminoacid residues form a β -sheet or α -helix
 - Prokaryotes:
 - Feature importance for pathogenic vs. non-pathogenic
 - Eukaryotes:
 - Features of a cancer cell vs normal cell

Thanks to:

- Prof. O. Kuipers
- Dr. S. van Hijum
- Rutger Brouwer
- Grayson Herman Kleine Carvalhal Bello de Miranda
- Department of Molecular Genetics

Questions

???

???

???

???

Datasources

Public databases

- DBTBS - Information about B. Subtilis <http://dbtbs.hgc.jp/>
- PubMed – information and links to Genetic information <http://www.ncbi.nlm.nih.gov/>

R project for statistical computing

- Rgui – R programming environment <http://www.r-project.org/>
- Bioconductor 2.0 – R package for life sciences <http://www.bioconductor.org/>
- Party – R package for randomForest
- randomForest – R package for randomForest
- Genesis – Microarray data analysis package
- GeneNet – Genetic network reconstruction <http://strimmerlab.org/software/genenet/>

Perl

- ActivePerl – Perl programming environment <http://www.perl.com/>

Matlab

- Matlab R2006a – Mathematical environment used for LVQ <http://www.mathworks.nl/>