# Protein names precisely peeled off free text

## Presentation by Warren Cheung

Sven Mika and Burkhard Rost

November 21, 2005

# Overview

Introduction

Methods

Validation

Conclusions

# Extract Protein Names from Free Text

- *Free Text*: scholarly papers, articles, reports — biomedical literature
  - natural language
- *Protein Names*: very similar to
  - descriptive terms: STAT (signal transducer and activator of transcription)
  - gene names (*myc-c* gene and myc-c protein)
  - cell cultures (CD4+-cells and CD4 protein)
  - similar to chemical compounds
  - not very standardized naming

# Definition of a Protein Name

- defines a *single* biological entity
- entity is composed of one or more amino acids
- removes ambiguities about genes, protein families, domains, etc.
- can look up name once identified in databases

# The Difficulty?

- protein databases exist — why do we need to extract names?
- associating text with *relevant protein* names
- identify new protein names from context
- automation — process previous work
- information is inaccessible to computers

# Motivation

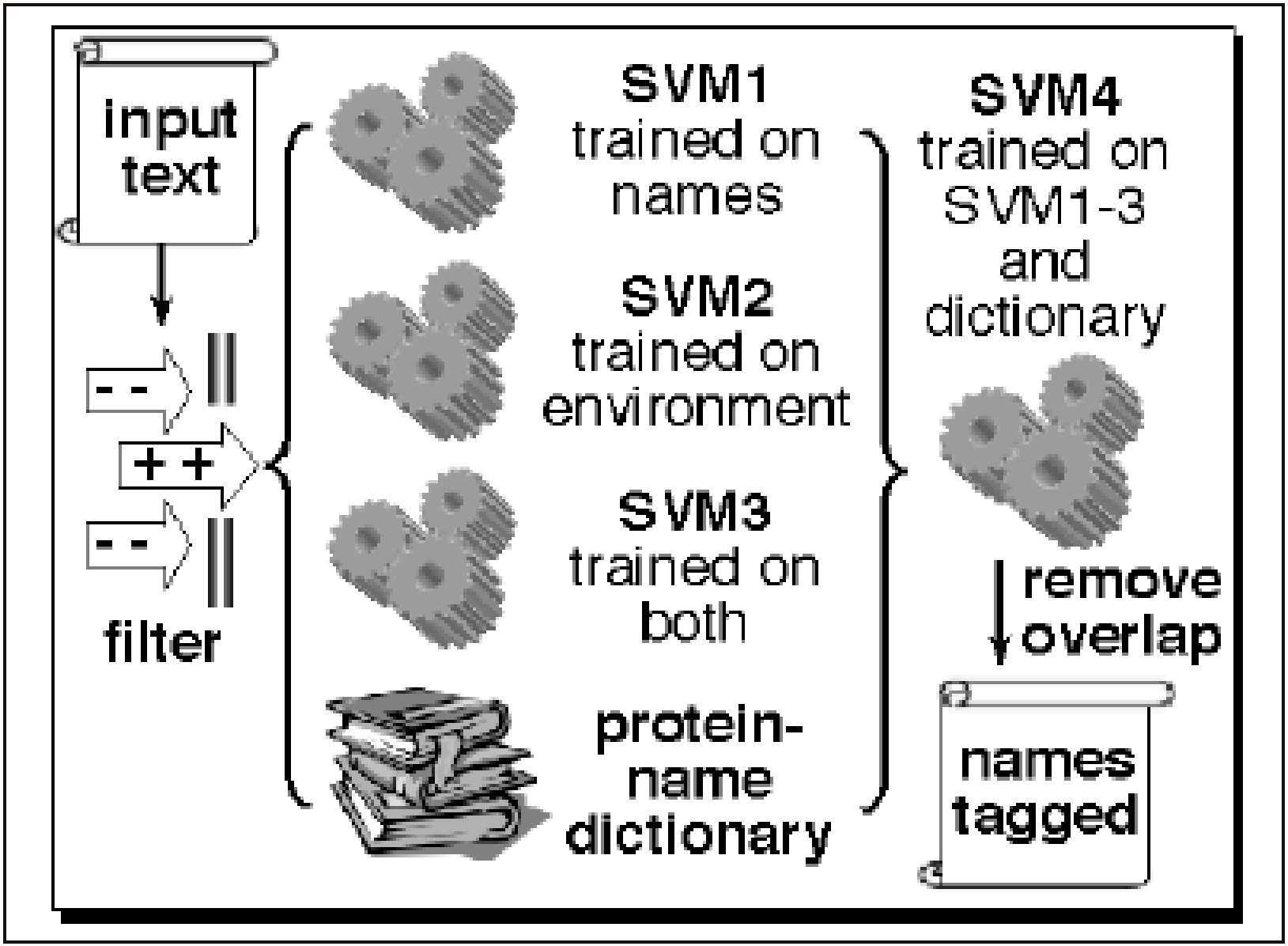
- core data in articles
  - quick test — removal can make text useless
- generate data for data mining
  - rapid extraction can be run on existing databases
- Authors developed *NLProt*

# Previous Work

- *rule-based*
  - specifically tailored to a a domain(SH3) 96% "effective"
  - general rules 67% — requires manual tweaking
- *dictionary-based*
  - identify possible protein names
  - use some kind of filter (e.g. similarity searching) to correctly identify actual protein names
  - limited to dictionary of terms
- *machine-learning*
  - Bayesian filter
  - hidden Markov models

Authors combine rules-based, dictionary-based and SVM classifiers

# NLProt


input text
- -
+ +
- -
filter
SVM1 trained on names
SVM2 trained on environment
SVM3 trained on both
protein-name dictionary
SVM4 trained on SVM1-3 and dictionary
remove overlap
names tagged

# Extracting Tokens

- separate punctuation marks with spaces
- spaces separate tokens
- slashes (/) and hyphens (-) do not
- *centre*: sliding window of 1—5 tokens
- *environment*: 4 tokens immediately before and after

| *environment 1* | | | | **centre** | | | | *environment 2* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *a* | *6-fold* | *decrease* | *in* | **high** | **mobility** | **group** | **protein** | *(* | *HMG* | *)* | *could* |
| *(1)* | *(2)* | *(3)* | *(4)* | *(A)* | *(B)* | *(C)* | *(E)* | *(5)* | *(6)* | *(7)* | *(8)* |

| *human* | *Rad51* | *amino* | *acid* | **residues** | **required** | *for* | *Rad52* | *binding* | . |
|---|---|---|---|---|---|---|---|---|---|
| *(1)* | *(2)* | *(3)* | *(4)* | *(A)* | *(E)* | *(5)* | *(6)* | *(7)* | *(8)* |

# Filtering

Use rules to filter out non-protein tokens

- dictionary of "common words"
  - Merriam-Webster dictionary (online version)
  - dictionary of medical terms
  - dictionary of minerals and formulas
- list of 130 4-letter chemical compound endings
- regular expressions (e.g. detect DNA sequences AGTGGC, author names)

| Rule | Text examples |
|---|---|
| Set of regular expressions | 16S, AGGTGGC, Ca2+, L214A, 26 kDa, mol/L, Asp-15 |
| Name is followed by 'cell(s)' or 'cyte(s)' | CD4+T lymphocytes, *Streptococcus mutans* cells |
| Name ending similar to chemical compound (list of 130 4-letter endings) | polypyrimidine, ether, ethanolamine |
| Name is in common-dictionary | interaction, factor, HIV-1, specific, leucocytes |
| Name seems to be an author | Miller *et al.*, Smith (2002) |
| Name is in parentheses following a filtered-out word | CD4+T lymphocytes (CD4TL), Inositol-3-phosphate (IP3) |
| Name is number followed by noun in plural form | four proteins, three factors |

# Text Matching

- all text converted to lowercase
- replace non-alphanumeric characters with spaces
- insert spaces between digits and letters

# Support Vector Machines (SVM)

- solve *two-class* classification problems
- map training input data into high-dimensional feature space
- find optimally separating hyperplane
- test new data against this hyperplane to classify
- *NLProt* combines four SVMs
- use publically available SVMlight package(`http://svmlight.joachims.org/`)

# Environment

- protein name itself
- surrounding words ( *"local"*)
- occurences elsewhere in the text (*"global"* )
- position unspecific — count occurences
- position specific — relative order of words

# SVMs 1-4

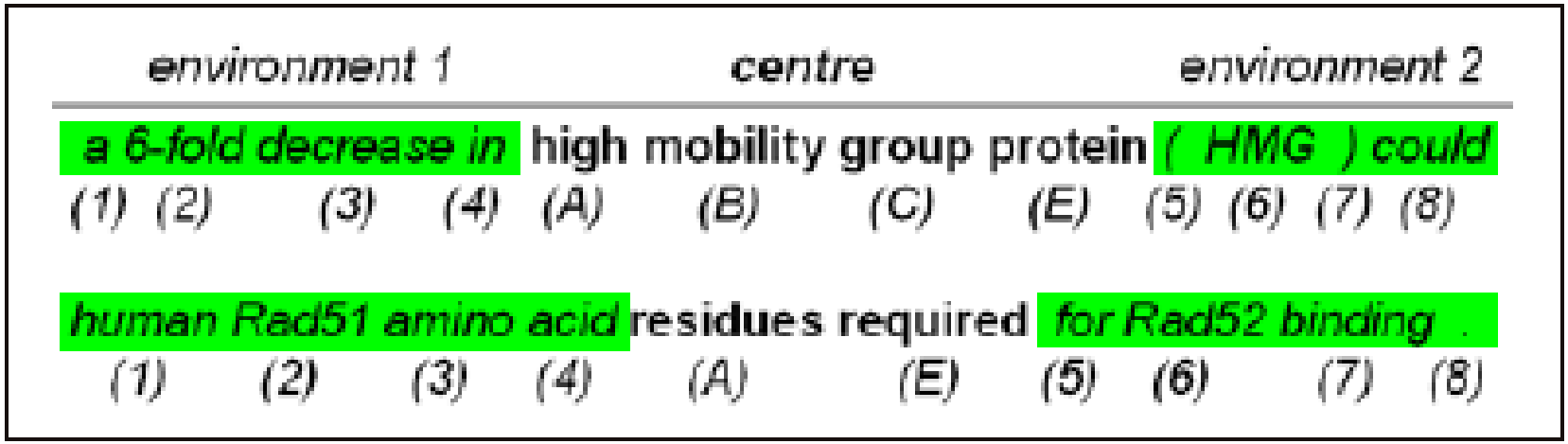
- SVM1: trained on the centre
- SVM2: trained on environment
- SVM3: trained on both overlap of centre and environment
- SVM4: input is SVM1-3 and dictionary score

# SVM1 (centre)

| environment 1 | centre | environment 2 |
|---|---|---|
| a 6-fold decrease in | high mobility group protein | ( HMG ) could |
| (1) (2) (3) (4) | (A) (B) (C) (E) | (5) (6) (7) (8) |
| human Rad51 amino acid | residues required | for Rad52 binding . |
| (1) (2) (3) (4) | (A) (E) | (5) (6) (7) (8) |

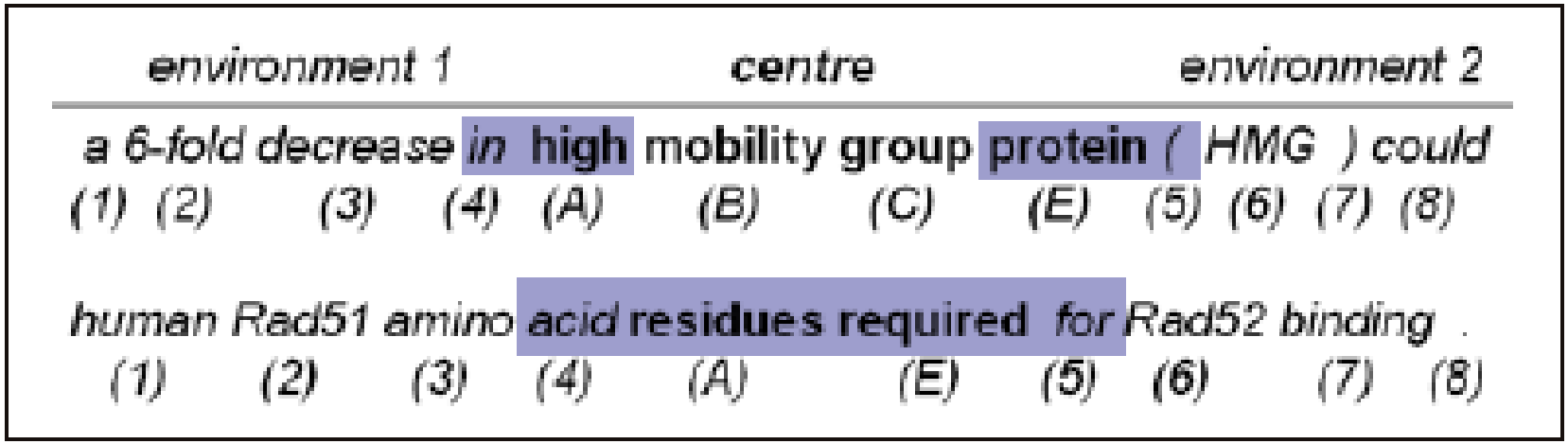
- get 3000 most commonly occurring centre tokens
- create 9000 component vector
- first 3000 components: occurence of common token in leftmost position
- 3001-6000: occurs at the rightmost position
- 6001-9000: occurs in the middle position

# SVM2 (environment tokens)




- get 3000 most commonly occurring environment tokens
- 12000 component feature vector
- first 3000 components: weight 0.75-0.25 if token occurs 2-4 tokens to the left of the centre
- 3001-6000: weight 1.0 if token occurs immediately to the left
- 6001-9000: weight 1.0 if token occurs immediately to the right
- 9001-12000: weight 0.75-0.25 if token occurs 2-4 tokens to the right

# SVM3 (overlap)




- get 3000 most commonly occurring tokens
- first 3000 components: token occurs immediately to the left of centre
- 3001-6000: token is leftmost of the centre
- 6001-9000: token is rightmost of the centre
- 9001-12000: token is immediately right of centre

# SVM4

- input is the output of SVM1-3
- also have a dictionary score
- protein names from SWISS-PROT
- if centre string matches, score is length of the string

# Validation

- 40-fold cross-validation for most results
- 200 articles — train SVM1-3 with 180, SVM4 with 15 and test with 5
- use harmonic mean of accuracy and coverage ($F$) as the measure
- *accuracy*: percentage of identified proteins that are true proteins
- *coverage*: percentage extracted of all the proteins names
- we want both of these to be as high as possible

# Results

| Method | Testing corpus | Number of protein names | Bias? | Accuracy (%) | Coverage (%) | *F* (%) |
|---|---|---|---|---|---|---|
| NLProt | Yapex | 1938 | Yes | 75 | 76 | 75 |
| NLProt | GENIA | 20 778 | Yes | 63 | 81 | 71 |
| NLProt | Recent166 | 1349 | Yes | 70 | 85 | 77 |
| NLProt | Yapex101 | 1938 | Yes | 76 | 78 | 77 |
| Yapex | Yapex101 | 1938 | Yes | | | 67 |
| NLProt | BioCreAtIvE | 11 871 | Yes | 73 | 75 | 74 |
| NLProt | Yapex | 1109 | No | 61 | 59 | 60 |
| NLProt | GENIA | 4104 | No | 48 | 60 | 53 |

# Bias

- reduce redundancy, overfitting
- performance on *novel* protein names
- remove names in training set/dictionary from testing set ($F = 60\%, F = 53\%$)
- remove only dictionary ($F = 72\%$)

# Contribution of Environment

- $F = 75\%$
- Using just SVM2 and dictionary $F = 69\%$
- Using just SVM1 and dictionary $F = 63\%$

# Conclusions

- automated tools can get a rough extraction (70%) of protein names
- supplement but cannot yet replace manual methods
- NLProt is competitive with the best existing methods