Mineração de Dados
Aula 4 – parte 2

Especialização em Ciência de Dados e suas Aplicações

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Text mining

- **Encontrar padrões interessantes em grandes volumes de bases de dados textuais**
  - **Interessante: desconhecido, não-trivial, potencialmente útil**

Palavras

Propriedades:

A frequência de palavras em um texto tendem a seguir uma lei de potência:

- **Pequeno número de palavras muito frequentes**
- **Número grande de palavras infrequentes**

Stop-words

De um ponto de vista não-linguístico não carregam informação:

- Usualmente são removidas para ajudar alguns métodos e melhorar o desempenho

Exemplos:

Inglês: A, About, Above...
Português: Estejam, elas, estou...

É interessante eliminar

Stemming

Processo de transformar uma palavra em sua forma normalizada (base ou *stem*)

Diferentes formas da mesma palavra pode ser problemático:

- **EX: Learns, learned, learning**

Stemming

Em Inglês:

Porter stemmer é um algoritmo amplamente usado

Exemplos de regras do Porter

ATIONAL -> ATE	relational -> relate
TIONAL -> TION	conditional -> condition
ENCI -> ENCE	valenci -> valence
ANCI -> ANCE	hesitanci -> hesitance
IZER -> IZE	digitizer -> digitize
ABLI -> ABLE	conformabli -> conformable
ALLI -> AL	radicalli -> radical
ENTLI -> ENT	differentli -> different
ELI -> E	vileli - > vile
OUSLI -> OUS	analogousli -> analogous

N-grams

- Conjunto de palavras coocorrendo dentro de uma determinada janela
- Ao calcular os n-grams normalmente movemos uma palavra para a frente (pode ser mais)

"The cow jumps over the moon".

Se n=2

- the cow
- cow jumps
- jumps over
- over the
- the moon

5 n-gramas

Se n=3

- the cow jumps
- cow jumps over
- jumps over the
- over the moon

4 n-gramas

Utilidade

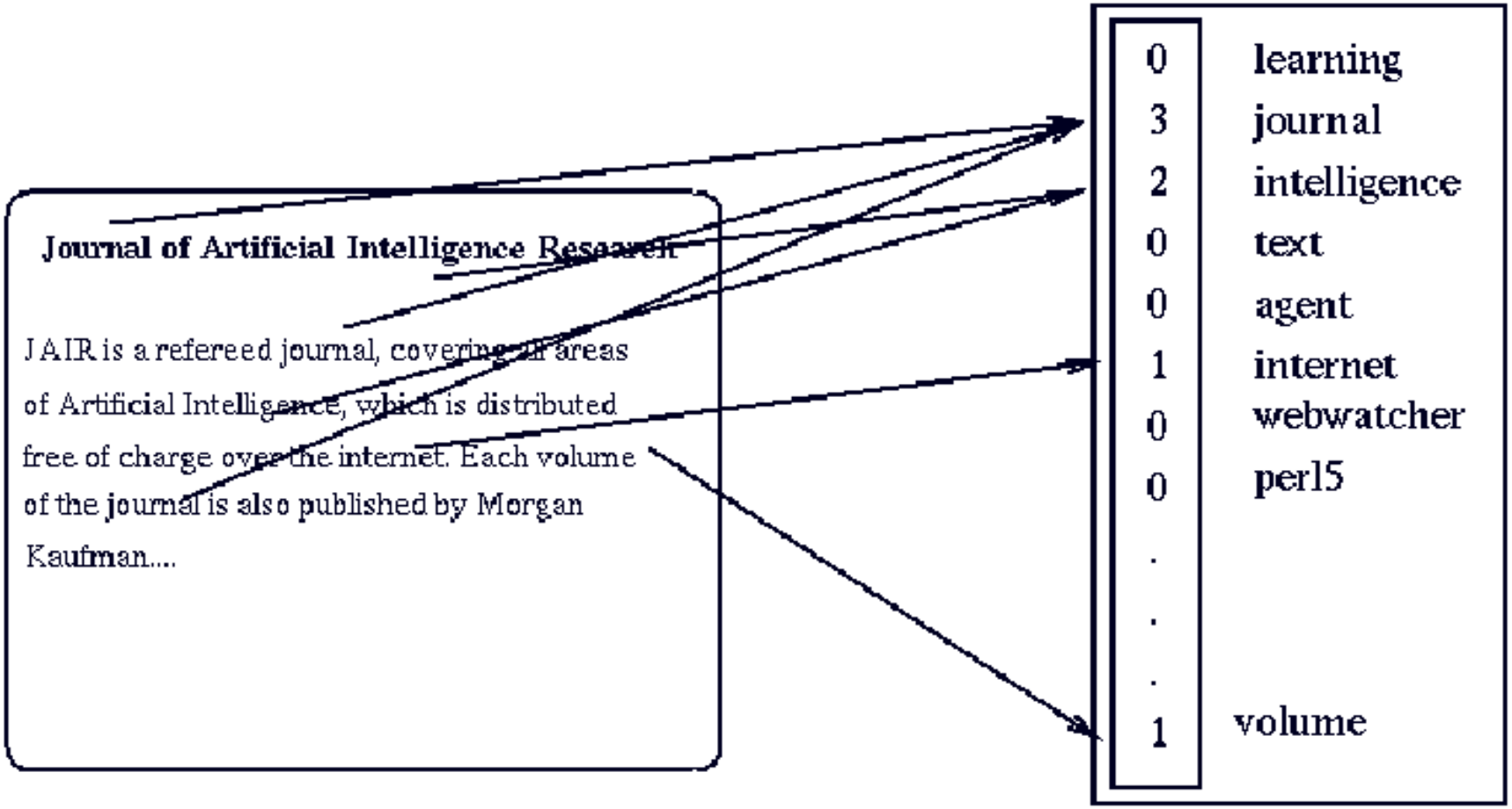
Quebra de palavras

Sugestão de próxima palavra....

https://www.microsoft.com/cognitive-services/en-us/web-language-model-api

Representação de documentos

Uma estratégia: Bag-of-words



Termos comuns, como “the”, vão ter frequência elevada

Representação de documentos

Uma estratégia: **Bag-of-words**

- **Palavras possuem um peso numérico**
- **Uma abordagem comum é TFIDF:**

$$tfidf(w) = tf . \log(\frac{N}{df(w)})$$

- Tf(w) – term frequency (number of word occurrences in a document)
- Df(w) – document frequency (number of documents containing the word)
- N – number of all documents
- Tfidf(w) – relative importance of the word in the document

The word is more important if it appears several times in a target document

The word is more important if it appears in less documents

Document-term matrix

Matriz que lista todas as ocorrências de palavras no corpus por documento.

Documentos são as linhas e os termos as colunas

D1 = "I like databases"
D2 = "I hate databases"

	I	like	hate	databases
D1	1	1	0	1
D2	1	0	1	1

Outros pesos poderiam ser usados, por exemplo, o tf-idf

Facilitadores

- NLTK - python
- TM - R

Separa sentenças e palavras

```python
# coding=utf-8

from nltk.tokenize import sent_tokenize

texto = "Este é um curso de Computação Social. Prof. Thiago DAINF. Olho d'água."

sentencas = sent_tokenize(texto)

for sent in sentencas:
    print sent

```

```python
# coding=utf-8

from nltk.tokenize import word_tokenize

texto = "Este é um curso de Computação Social. Prof. Thiago DAINF. Olho d'água."

palavras = word_tokenize(texto)

for palavra in palavras:
    print palavra
```

Separa sentenças e palavras

```
1  # coding=utf-8
2
3  from nltk.tokenize impo
4
5  texto = "Este é um curs
6
7  sentencas = sent_tokeni
8
9  for sent in sentencas:
10     print sent
11
```

```
Este
é
um
curso
de
Computação
Social
.
Prof.
Thiago
DAINF
.
Olho
d'água
.
```

```
1  # coding=utf-8
2
3  from nltk.tokenize impor
4
5  texto = "Este é um curso
6
7  palavras = word_tokenize
8
9  for palavra in palavras:
10     print palavra
```

```
Este é um curso de Computação Social.
Prof. Thiago DAINF.
Olho d'água.
```

Separando palavras

In Curitiba I took my hat off. But I can't put it back on.

TreebankWordTokenizer

1. In | Curitiba | I | took | my | hat | off | .
2. But | I | ca | n't | put | it | back | on | .

WordPunctTokenizer

1. In | Curitiba | I | took | my | hat | off | .
2. But | I | can | ' | t | put | it | back | on | .

PunktWordTokenizer

1. In | Curitiba | I | took | my | hat | off.
2. But | I | can | 't | put | it | back | on.

WhitespaceTokenizer

1. In | Curitiba | I | took | my | hat | off.
2. But | I | can't | put | it | back | on.

pattern

1. In | Curitiba | I | took | my | hat | off | .
2. But | I | ca | n't | put | it | back | on | .

Derivado de: http://text-processing.com/demo/tokenize/

Wordclouds


valley
straight
prodigious
transform
owners
mighty
justice
mountains
hampshire
sweet
still
oppression
curvaceous
colorado
jews
sisters
liberty
old
glory
back
children
every
governor
slave
catholics
must
brothers
nullification
beautiful
brotherhood
sweltering
though
village
speed
hills
hew
georgia
girls
mountain
rooted
flesh
red
jangling
stone
country
men
one
free
former
injustice
plain creed
nation
black
white
molehill
deeply
stand
heat
become
pilgrim
god
freedom
even
transformed
rough
mountainside
hope
low
spiritual
tennessee
pray
snowcapped
racists
discords
sons
land
lord
live
will
crooked
see
hands
true
boys
happens
lips
little
faith
today
slaves
great
revealed
interposition
dream
let
sing
gentiles
dripping
alabama
made
ring
last
despair
table
places
color
work
almighty
skin
day
died
new
city
jail
struggle
created
four
difficulties
alleghenies
able
character
pride
tis
thee
allow
state
shall
hilltops
hold
fathers
join
california
protestants
together
exalted
america
sit
american
rockies
york
right
tomorrow
rise
meaning
hill
words
content
equal
mississippi
face
judged
south
lookout
hamlet
heightening
oasis
vicious
selfevident
knowing
negro
pennsylvania
symphony
slopes
thank
truths

Identificação de tópicos em textos

Modelos para tópicos

Métodos para descobrir temas (tópicos) de uma coleção

- **Ex: Livros, jornais, etc…**

Anota a coleção de acordo com os temas descobertos

Usa as anotações para organizar, procurar, agrupar…

Será discutido o Latent Dirichlet Allocation - LDA

Principais suposições do LDA

Documentos possuem múltiplos tópicos (mas tipicamente não muitos)

LDA é um modelo probabilístico com um processo generativo correspondente

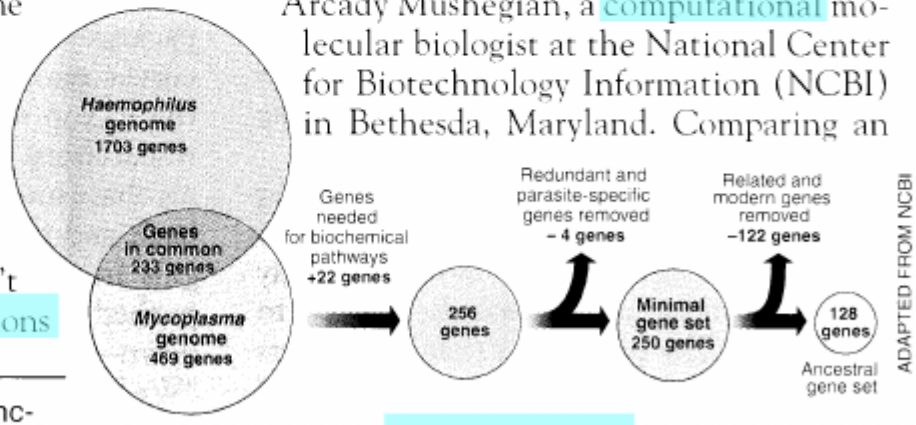
- **É assumido que cada documento é gerado por esse processo**
- **Um tópico é uma distribuição sobre um vocabulário fixo**
- **É assumido que os tópicos são gerados primeiro, antes do documento**
- **Somente o número de tópicos é especificado previamente**

Principais suposições do LDA

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

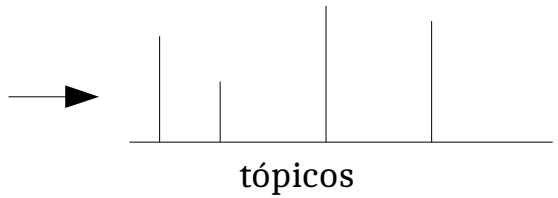
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

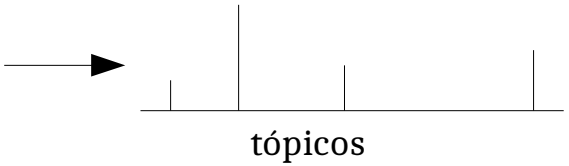
Documentos exibem múltiplos tópicos

Principais suposições do LDA

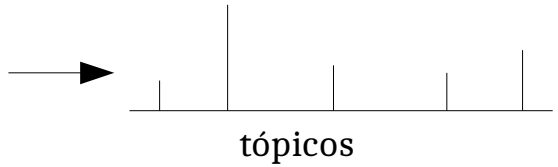
Documento 1



Documento 2



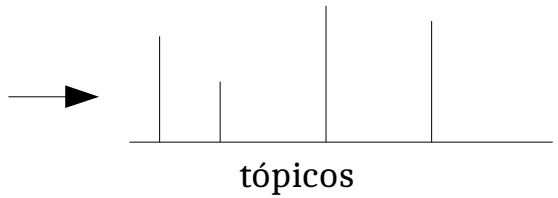
Documento 3



...

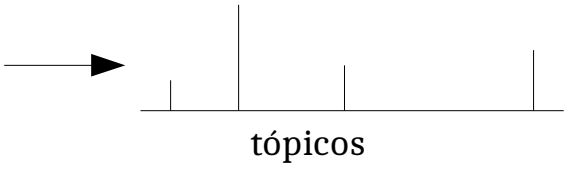
Principais suposições do LDA

Documento 1



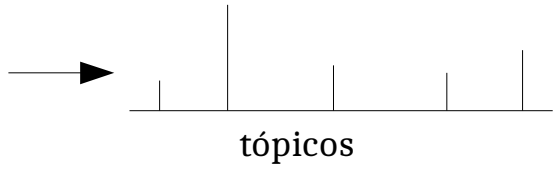
tópicos

Documento 2



tópicos

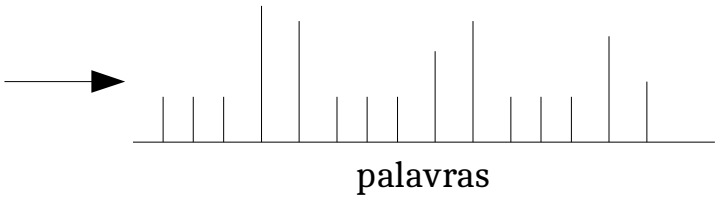
Documento 3



tópicos

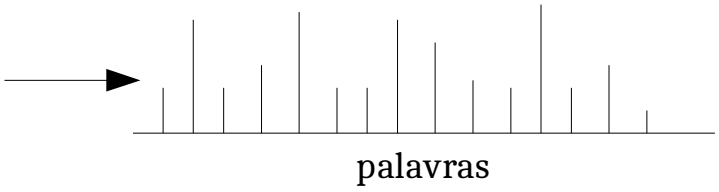
...

Tópico 1



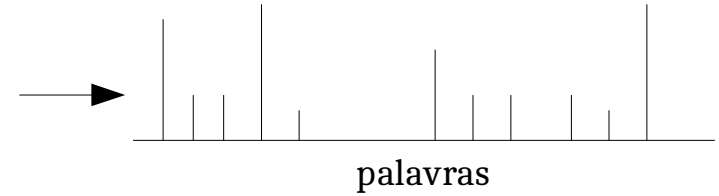
palavras

Tópico 1



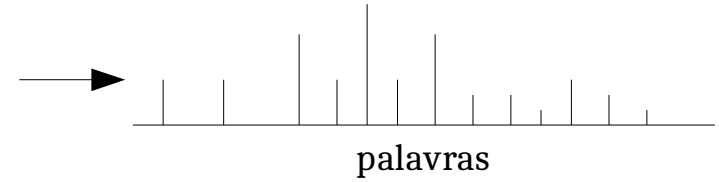
palavras

Tópico 1



palavras

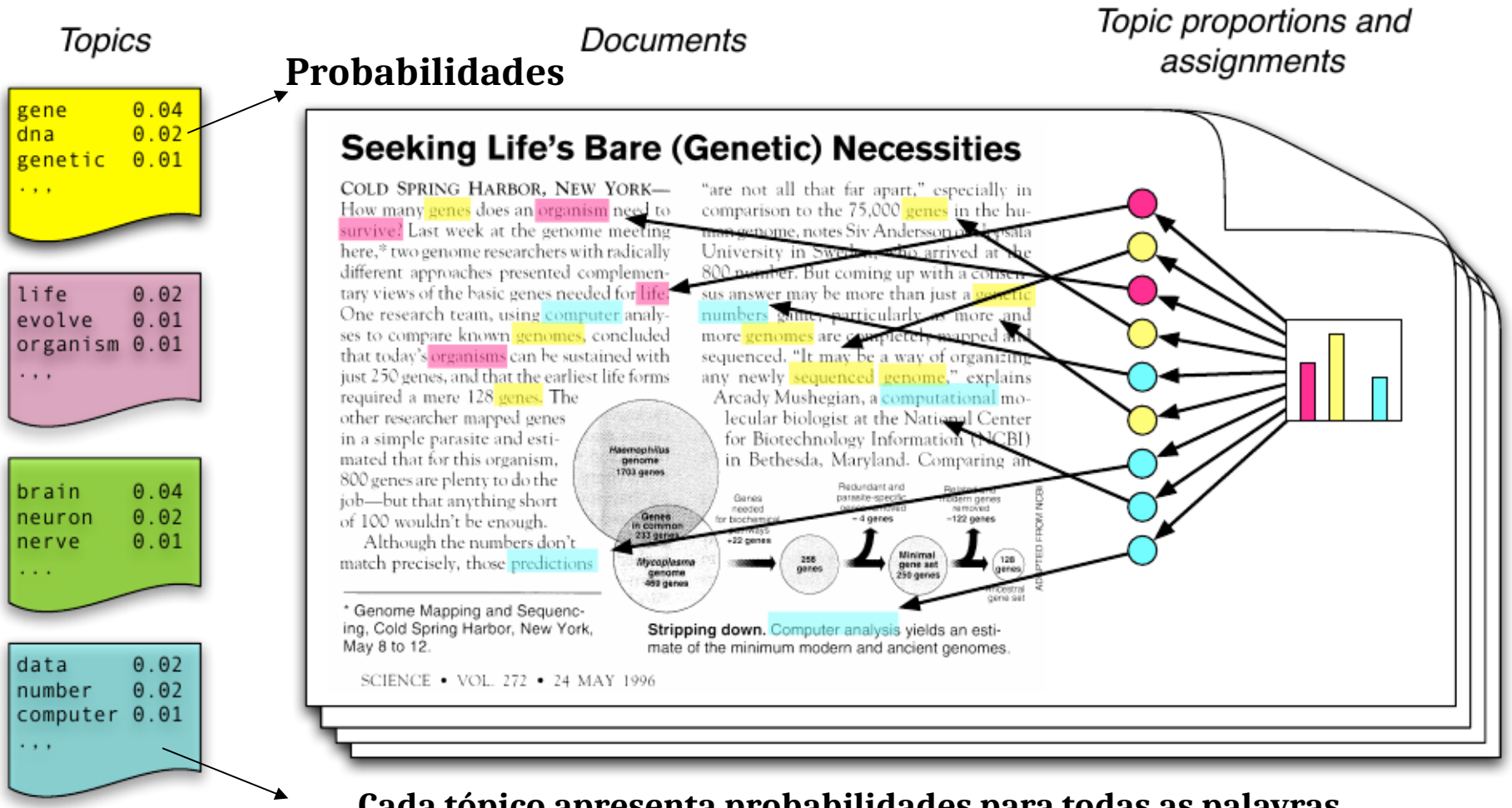
Tópico 1



palavras

...

Principais suposições do LDA



Cada tópico apresenta probabilidades para todas as palavras.
Uma mesma palavra pode ter probabilidade alta para diferentes tópicos

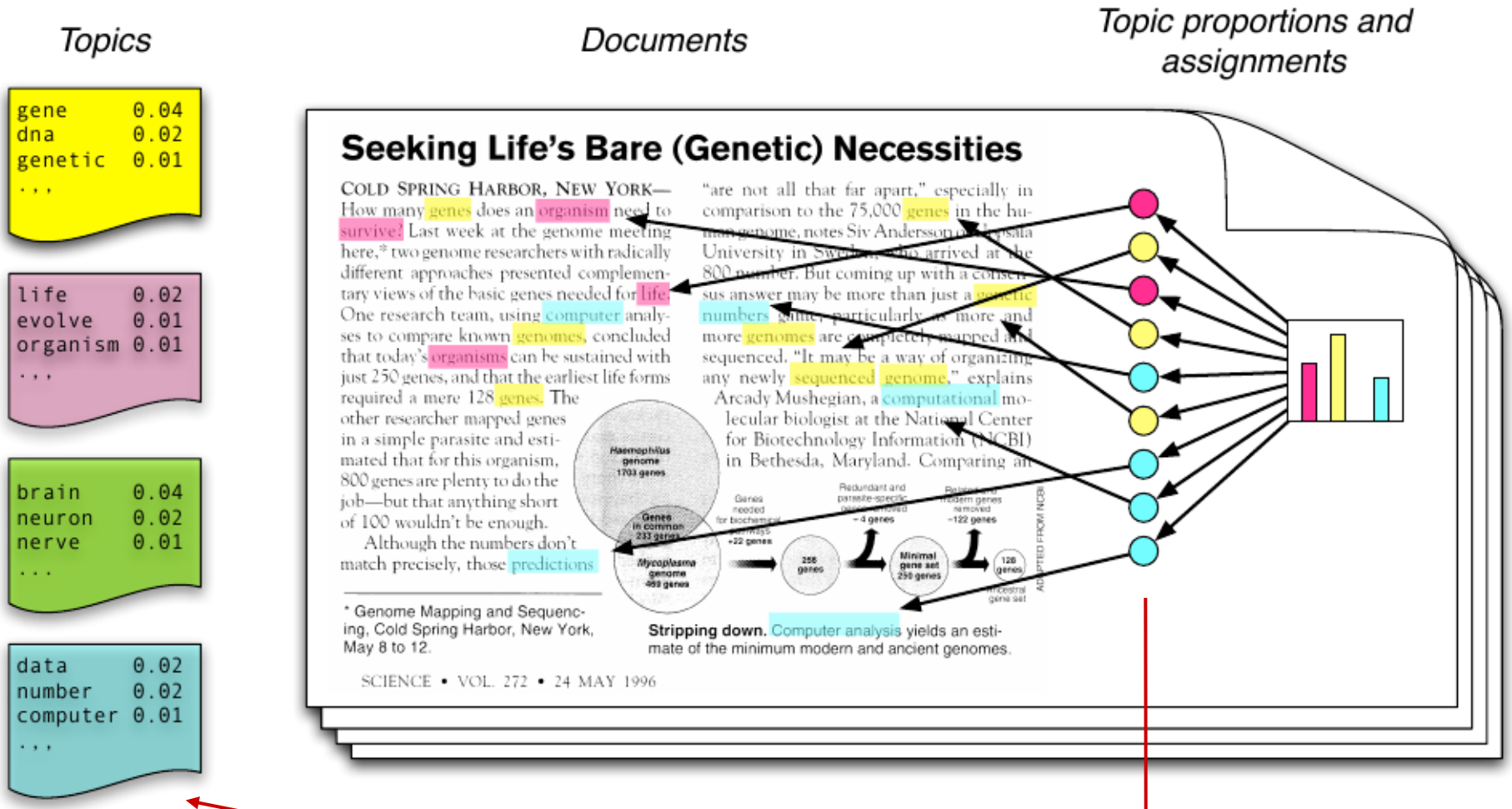
Modelo generativo

Cada tópico é uma distribuição sobre palavras (vocabulário fixo)

Cada documento é uma mistura aleatória de tópicos

Cada palavra é tirada desses tópicos

Principais suposições do LDA



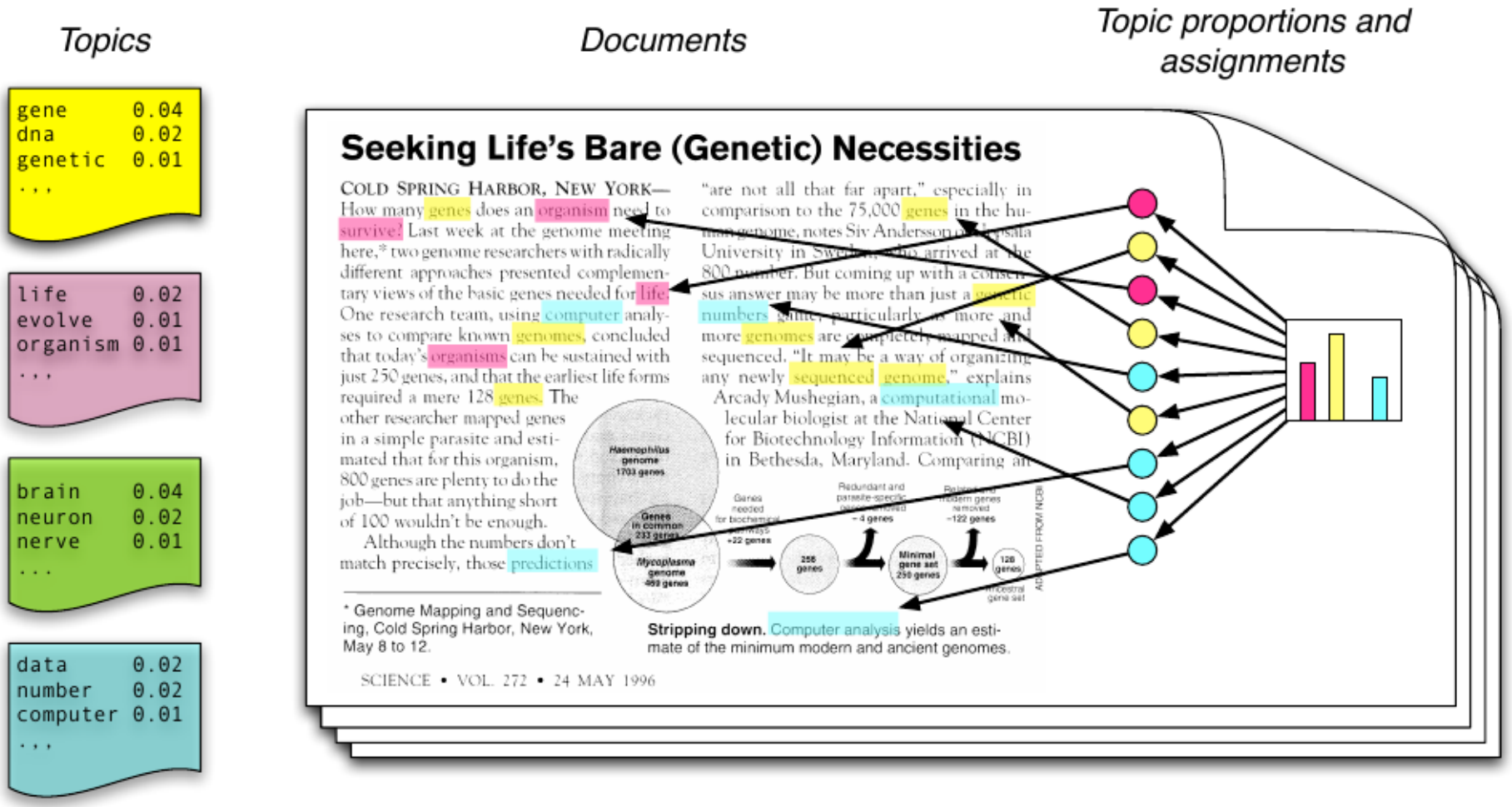
2

Em seguida é escolhida uma palavra de acordo com a distribuição correspondente

1

Distribuição de distribuições (Dirichlet)
Primeiro é selecionado um tópico

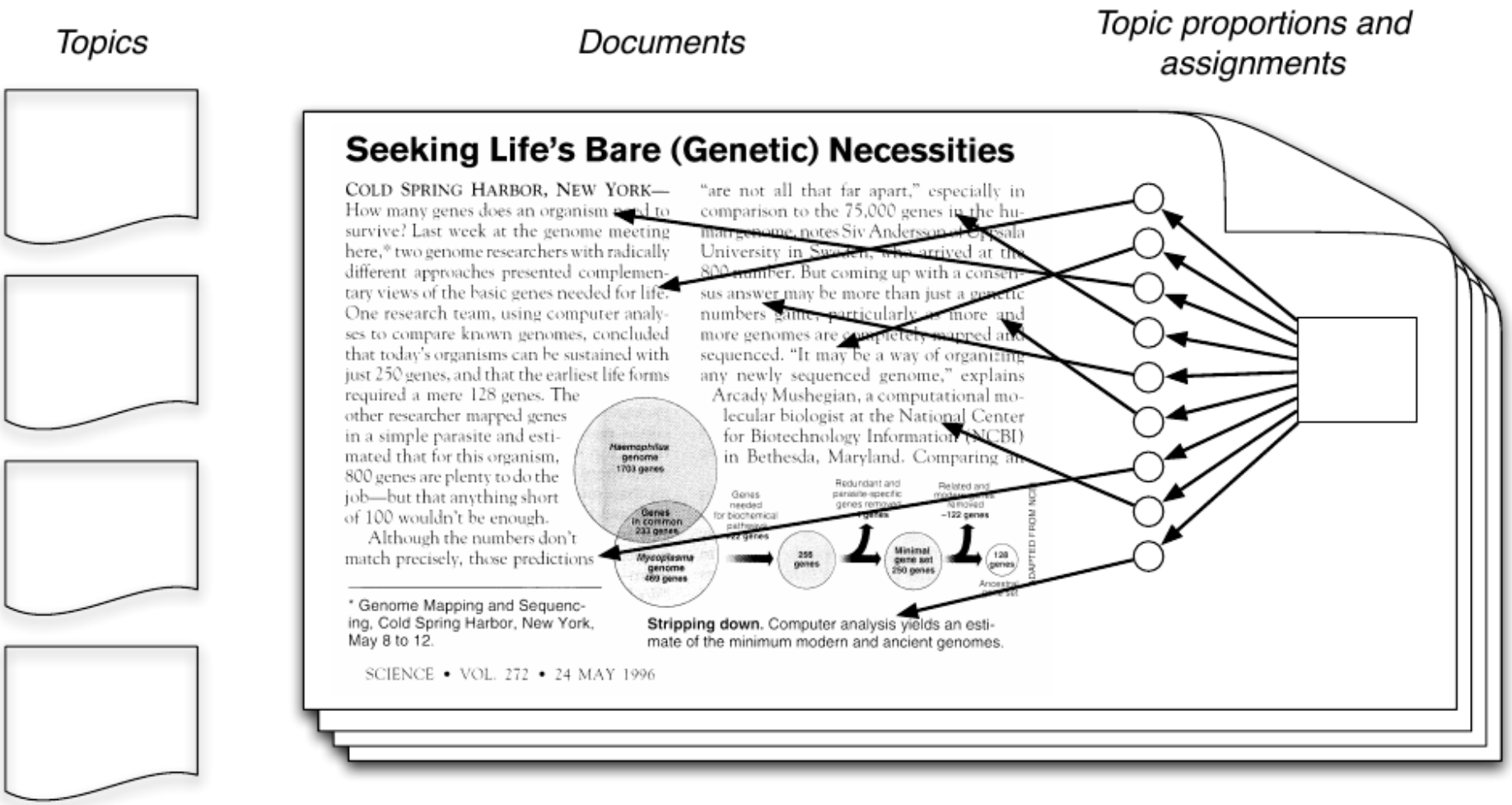
Principais suposições do LDA



Com esse processo generativo as ordem das palavras não importam (aleatórias)

Apesar de documentos não serem construídos assim, isso serve para sabermos os tópicos dos documentos.

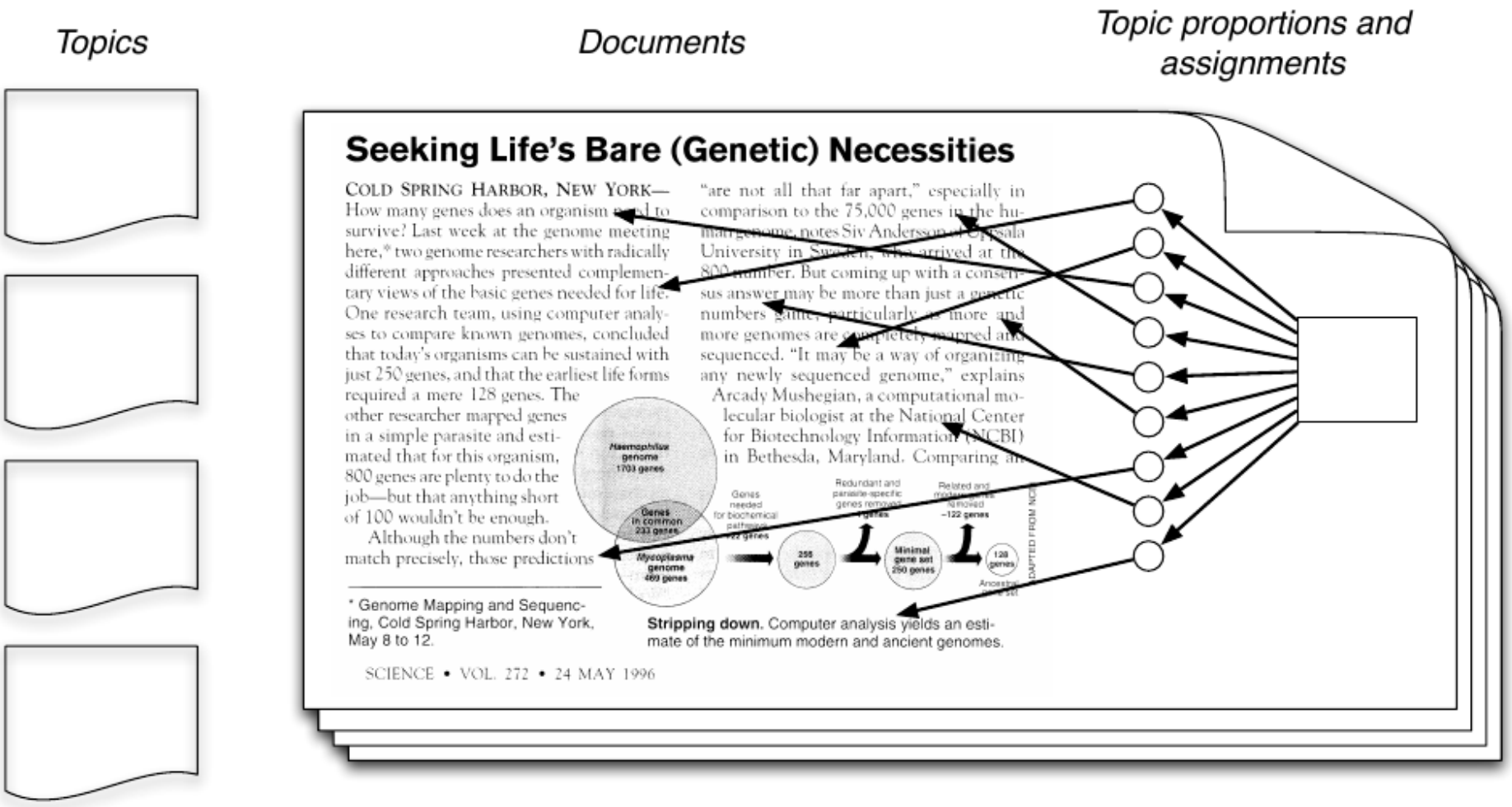
Principais suposições do LDA



Somente documentos são observados **(só o que temos)**

Tópicos, misturas (mixtures), etc. são todos escondidos e precisam ser aprendidos/preditos dos dados **(nosso objetivo)**

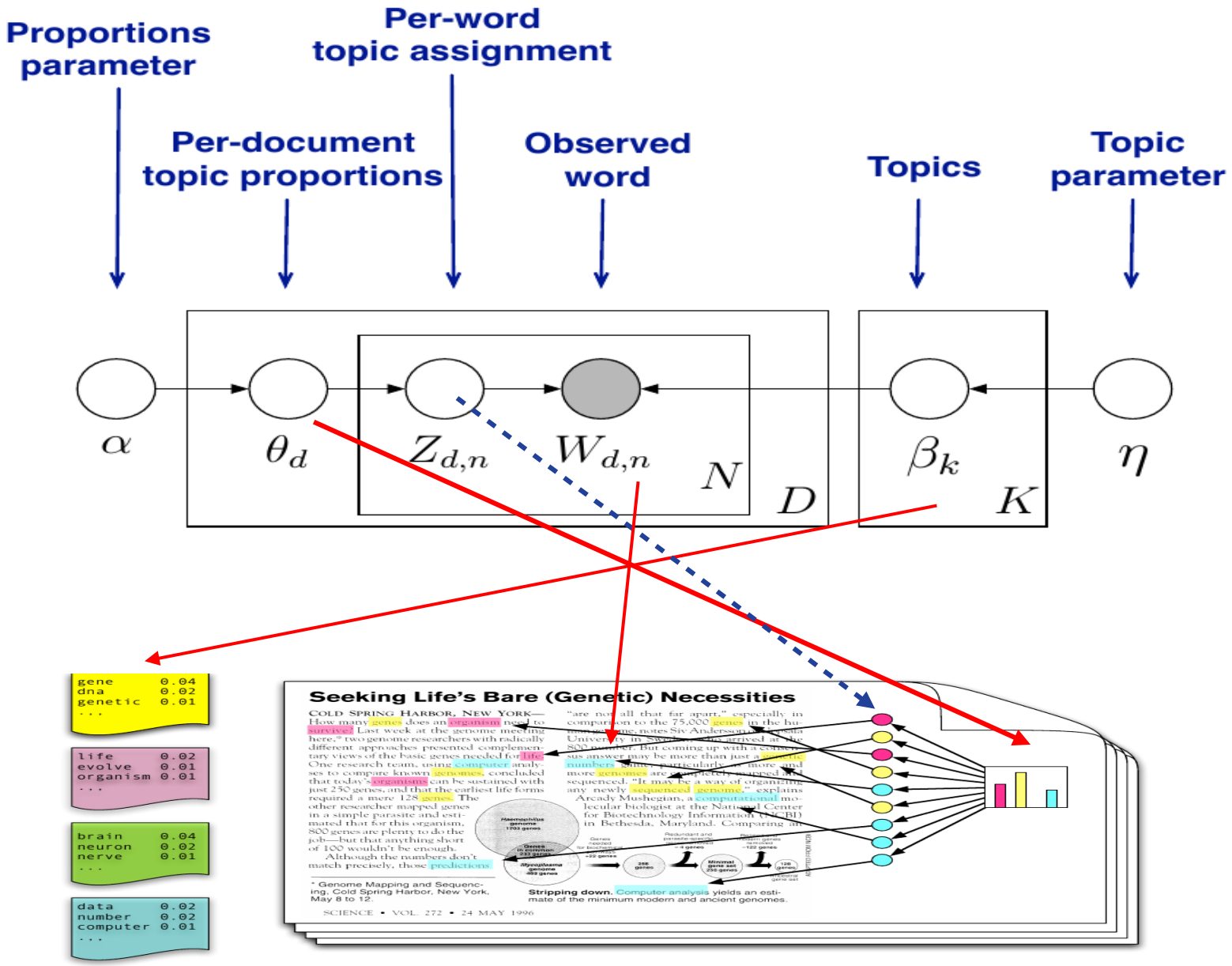
Principais suposições do LDA



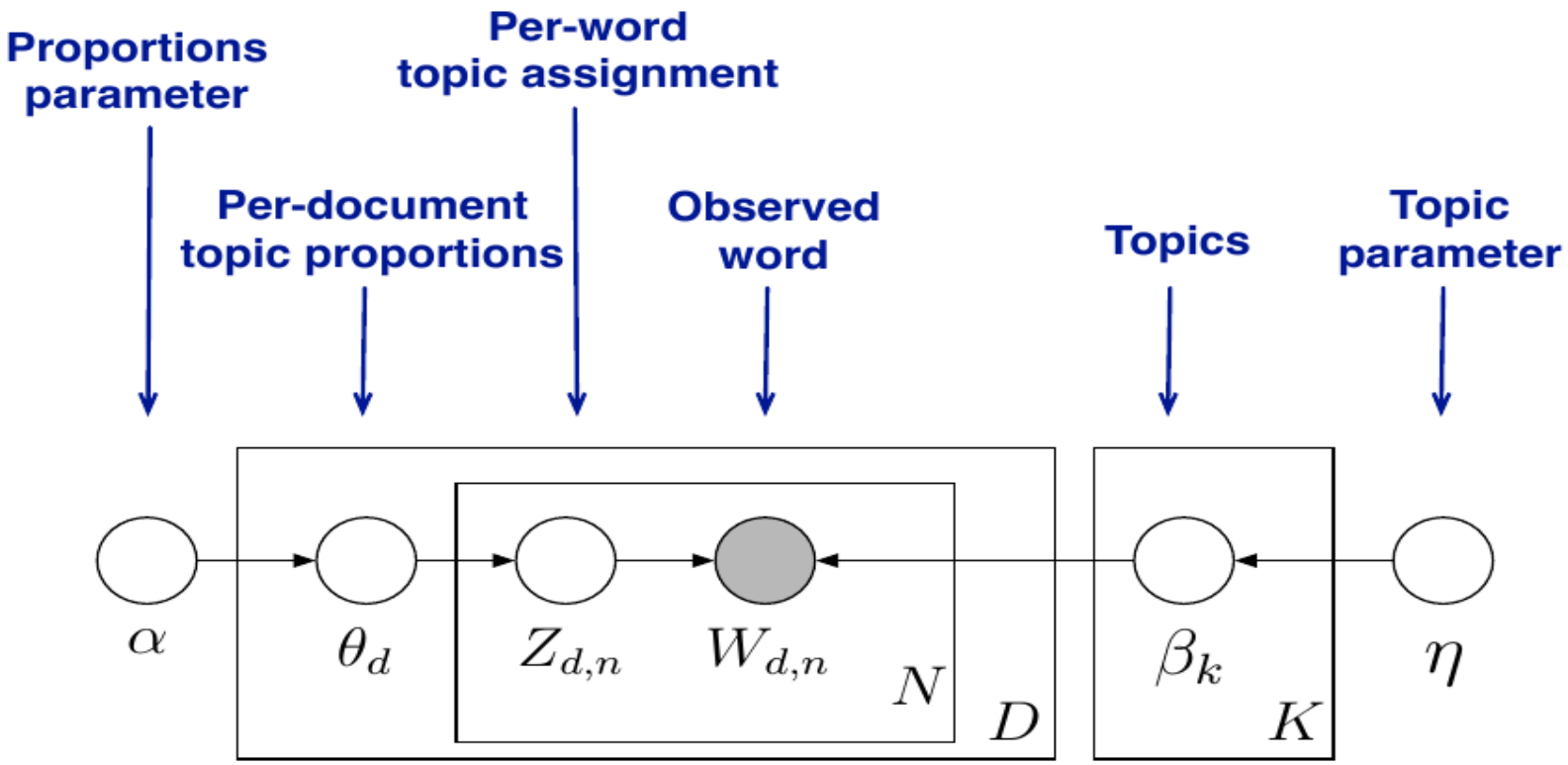
Ou seja, computar sua distribuição condicionada aos documentos

p(tópicos, proporção, atribuições | documentos)

Modelo gráfico para o LDA



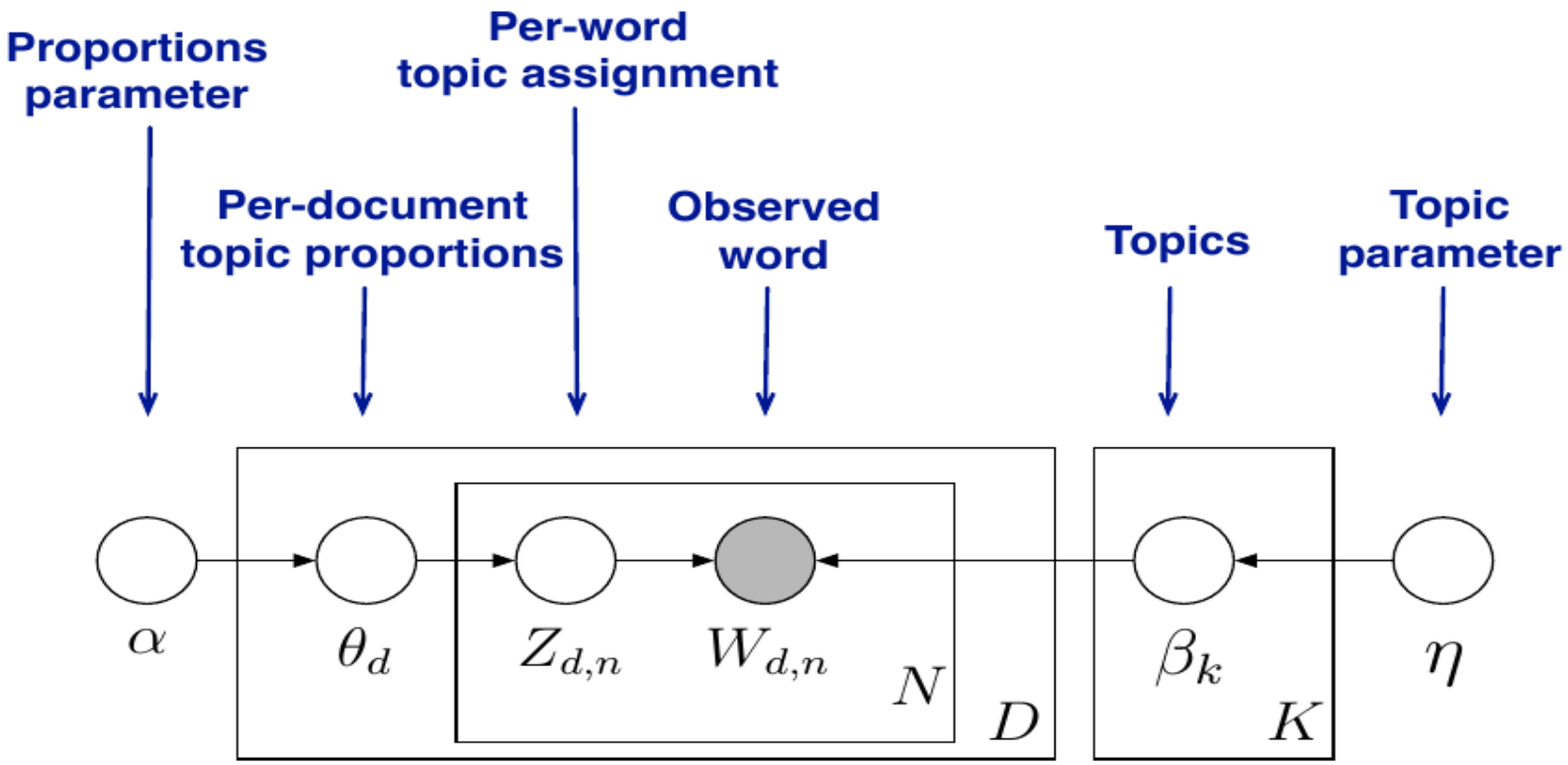
Modelo gráfico para o LDA



A única variável observada são as palavras nos documentos

O **tópico para cada palavra**, a **distribuição sobre os tópicos para cada documento** e a **distribuição de palavras por tópico** são **variáveis latentes** nesse modelo

Modelo gráfico para o LDA

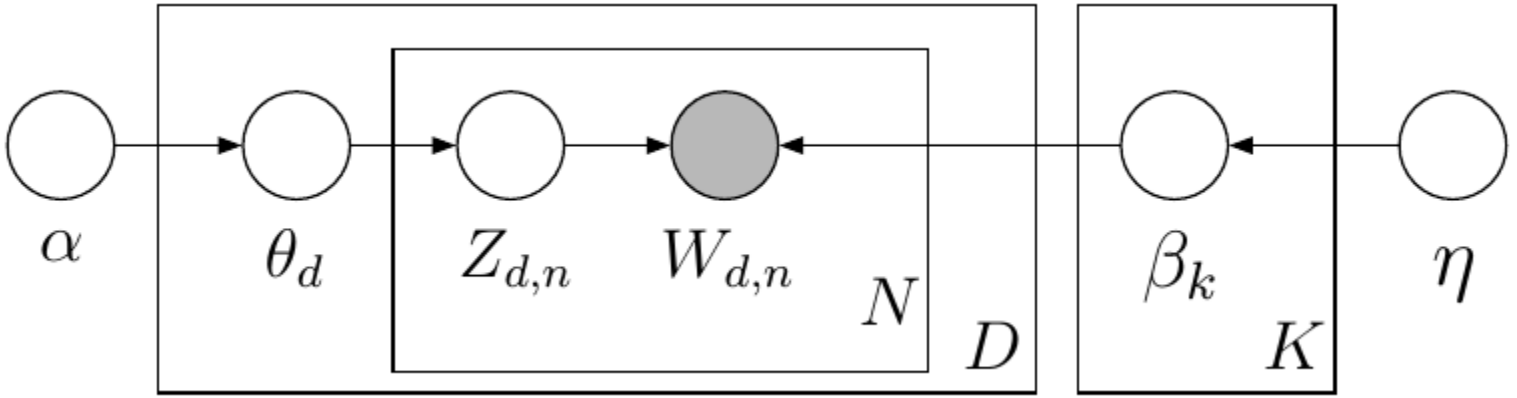


A única variável observada são as palavras nos documentos

Não observadas, mas são inferidas pela variável observada

O **tópico para cada palavra**, a **cada documento** e a **distribuição de palavras por tópico** são **variáveis latentes** nesse modelo

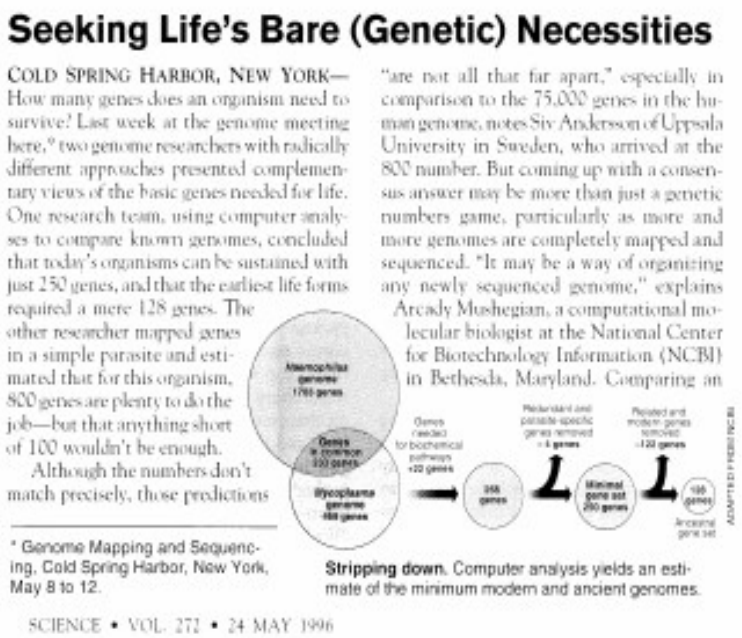
Modelo gráfico para o LDA



Algoritmos para a inferência aproximada da probabilidade a posteriori:

- Mean field variational methods (Blei et al., 2001, 2003)
- Expectation propagation (Minka and Lafferty, 2002)
- Collapsed Gibbs sampling (Griffiths and Steyvers, 2002)
- Collapsed variational inference (Teh et al., 2006)
- Online variational inference (Hoffman et al., 2010)

Exemplo de inferência



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

ADAPTED FROM NCBI

SCIENCE • VOL. 272 • 24 MAY 1996

- **Dados:** Coleção de artigos da Science de 1990–2000 (após OCR)
  - **17K documentos**
  - **11M palavras**
  - **20K termos únicos (*stop words* e palavras raras removidas)**

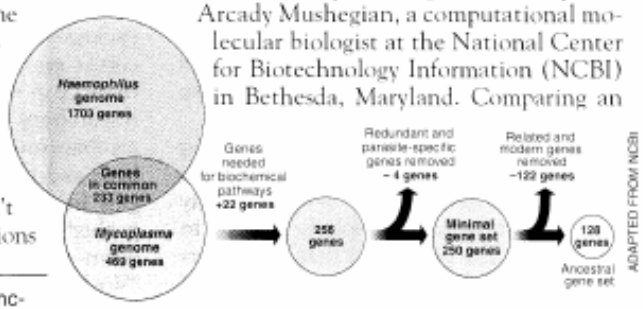
- **Modelo:** 100-tópicos em um modelo LDA usando variational inference.

Exemplo de inferência

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

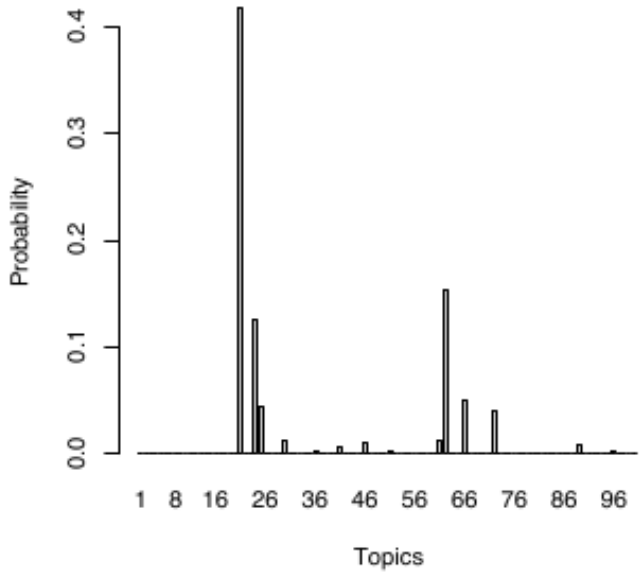
Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996



Proporção de tópicos que apareceram nesse documento (alguns possuem probabilidades mais altas)

Exemplo de inferência

human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

- 4 tópicos com maiores probabilidades
- Mostrando as 15 palavras com maior probabilidade

Exemplo de inferência

Chaotic Beetles

Charles Godfray and Michael Hassell

Ecologists have known since the pioneering work of May in the mid-1970s (*1*) that the population dynamics of animals and plants can be exceedingly complex. This complexity arises from two sources: The tangled web of interactions that constitute any natural community provide a myriad of different pathways for species to interact, both directly and indirectly. And even in isolated populations the nonlinear feedback processes present in all natural populations can result in complex dynamic behavior. Natural populations can show persistent oscillatory dynamics and chaos, the latter characterized by extreme sensitivity to initial conditions. If such chaotic dynamics were common in nature, then this would have important ramifications for the management and conservation of natural resources. On page 389 of this issue, Costantino *et al.* (*2*) provide the most convincing evidence to date of complex dynamics and chaos in a biological population—of the flour beetle, *Tribolium castaneum* (see figure).

The authors are in the Department of Biology, Imperial College at Silwood Park, Ascot, Berks, SL5 7PZ UK. E-mail: m.hassell@ic.ac.uk

It has proven extremely difficult to demonstrate complex dynamics in populations in the field. By its very nature, a chaotically fluctuating population will superficially resemble a stable or cyclic population buffeted by the normal random perturbations experienced by all species. Given a long enough time series, diagnostic tools from nonlinear mathematics can be used to identify the telltale signatures of chaos. In phase space, chaotic trajectories come to lie on "strange attractors," curious geometric objects with fractal structure and hence noninteger dimension. As they move over the surface of the attractor, sets of adjacent trajectories are pulled apart, then stretched and folded, so that it becomes impossible to predict exact population densities into the future. The strength of the mixing that gives rise to the extreme sensitivity to initial conditions can be measured mathematically estimating the Liapunov exponent, which is positive for chaotic dynamics and nonpositive otherwise. There have been many attempts to estimate attractor dimension and Liapunov exponents from time series data, and some candidate chaotic population have been identified (some insects, rodents, and most convincingly, human childhood diseases), but the statistical difficulties preclude any broad generalization (*3*).

Cannibalism and chaos. The flour beetle, *Tribolium castaneum*, exhibits chaotic population dynamics when the amount of cannibalism is altered in a mathematical model.

An alternative approach is to parameterize population models with data from natural populations and then compare their predictions with the dynamics in the field. This technique has been gaining popularity in recent years, helped by statistical advances in parameter estimation. Good ex-

Exemplo de inferência

problem	model	selection	species
problems	rate	male	forest
mathematical	constant	males	ecology
number	distribution	females	fish
new	time	sex	ecological
mathematics	number	species	conservation
university	size	female	diversity
two	values	evolution	population
first	value	populations	natural
numbers	average	population	ecosystems
work	rates	sexual	populations
time	data	behavior	endangered
mathematicians	density	evolutionary	tropical
chaos	measured	genetic	forests
chaotic	models	reproductive	ecosystem

Exemplo de inferência

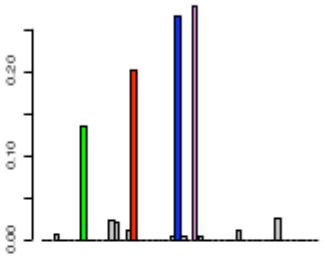
Chance and Statistical Significance in Protein and DNA Sequence Analysis

Samuel Karlin and Volker Brendel

Top words from the top topics (by term score)

sequence	measured	residues	computer
region	average	binding	methods
pcr	range	domains	number
identified	values	helix	two
fragments	different	cys	principle
two	size	regions	design
genes	three	structure	access
three	calculated	terminus	processing
cdna	two	terminal	advantage
analysis	low	site	important

Expected topic proportions



Usando tópicos para encontrar documentos similares (agrupamento)

Abstract with the most likely topic assignments

Statistical approaches help in the determination of significant configurations in protein and nucleic acid sequence data. Three recent statistical methods are discussed: (i) score-based sequence analysis that provides a means for characterizing anomalies in local sequence text and for evaluating sequence comparisons; (ii) quantile distributions of amino acid usage that reveal general compositional biases in proteins and evolutionary relations; and (iii) *r*-scan statistics that can be applied to the analysis of spacings of sequence markers.

Top Ten Similar Documents

Exhaustive Matching of the Entire Protein Sequence Database
How Big Is the Universe of Exons?
Counting and Discounting the Universe of Exons
Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment
Ancient Conserved Regions in New Gene Sequences and the Protein Databases
A Method to Identify Protein Sequences that Fold into a Known Three- Dimensional Structure
Testing the Exon Theory of Genes: The Evidence from Protein Structure
Predicting Coiled Coils from Protein Sequences
Genome Sequence of the Nematode C. elegans: A Platform for Investigating Biology

Ferramentas

TM – R
Topicmodel – R

NLTK – python
Gensim – python
(https://radimrehurek.com/gensim/)

Exemplo

_TesteLDA.r