LingPy – A Python Library for Quantitative Tasks in Historical Linguistics

Johann-Mattis List*

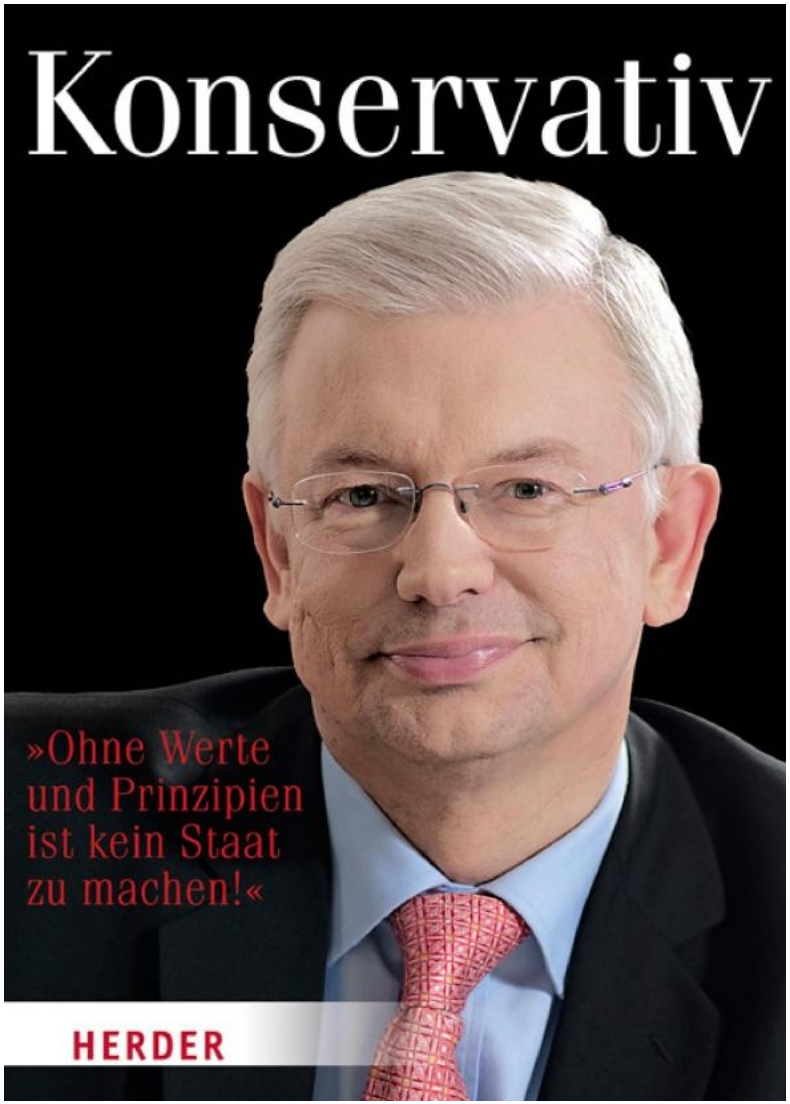
*Institut für Romanistik II
Heinrich Heine University Düsseldorf

19. Juli 2012

Introductory Remarks

What the *** is historical linguistics?

Introductory Remarks



Introductory Remarks

Konservativ kommt nicht von Konserve!

Angela Merkel (10. April 2000)

Introductory Remarks

konservativ

Introductory Remarks

konservativ

Part of Speech	adjective
frequency	moderate

Introductory Remarks

konservativ

Part of Speech	adjective
frequency	moderate

meaning	“sticking to the traditional”
synonyms	“rückschrittlich”, “antiquiert”, “rückständig”, “unzeitgemäß”
pronunciation	kɔnzɛrvatiːf

Introductory Remarks

konservativ

Part of Speech	adjective
frequency	moderate
meaning	“sticking to the traditional”
synonyms	“rückschrittlich”, “antiquiert”, “rückständig”, “unzeitgemäß”
pronunciation	kɔnzɛrvatiːf
origin	English *conservative*

Introductory Remarks

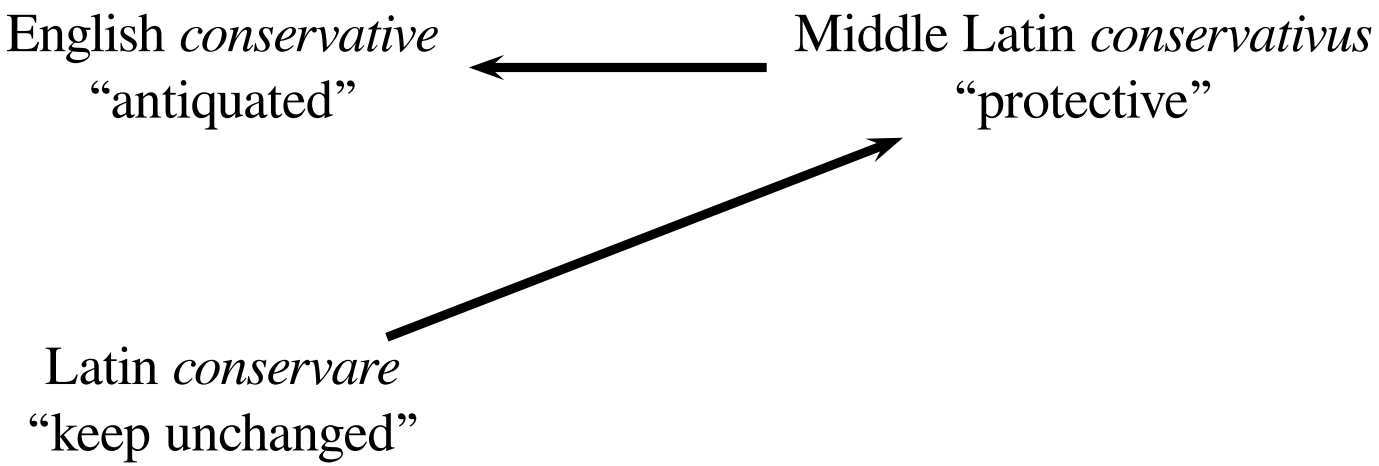
English *conservative*
"antiquated"

Introductory Remarks

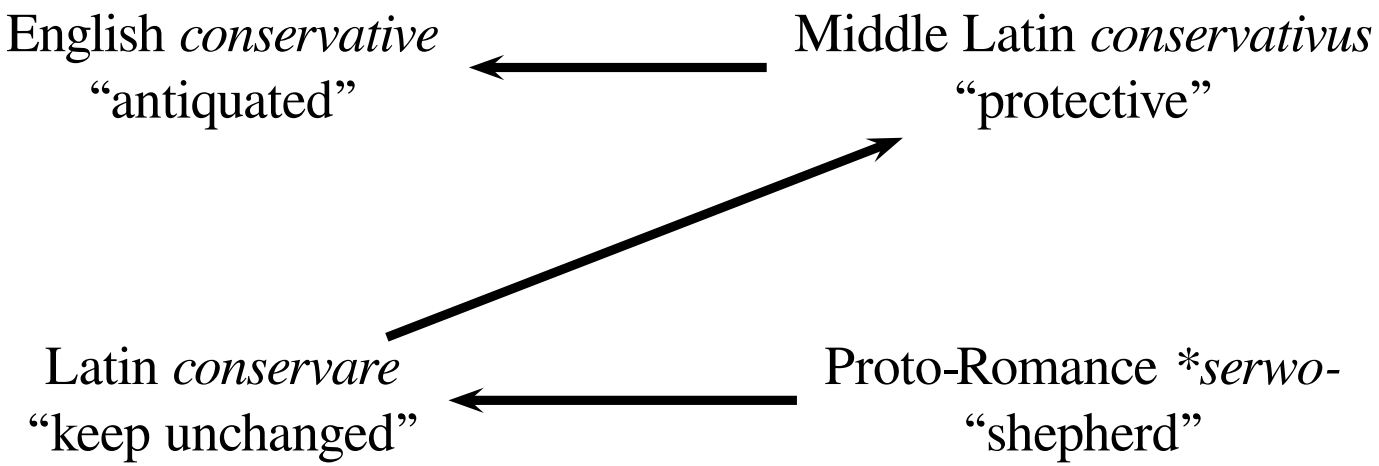
English *conservative*
"antiquated"

Middle Latin *conservativus*
"protective"

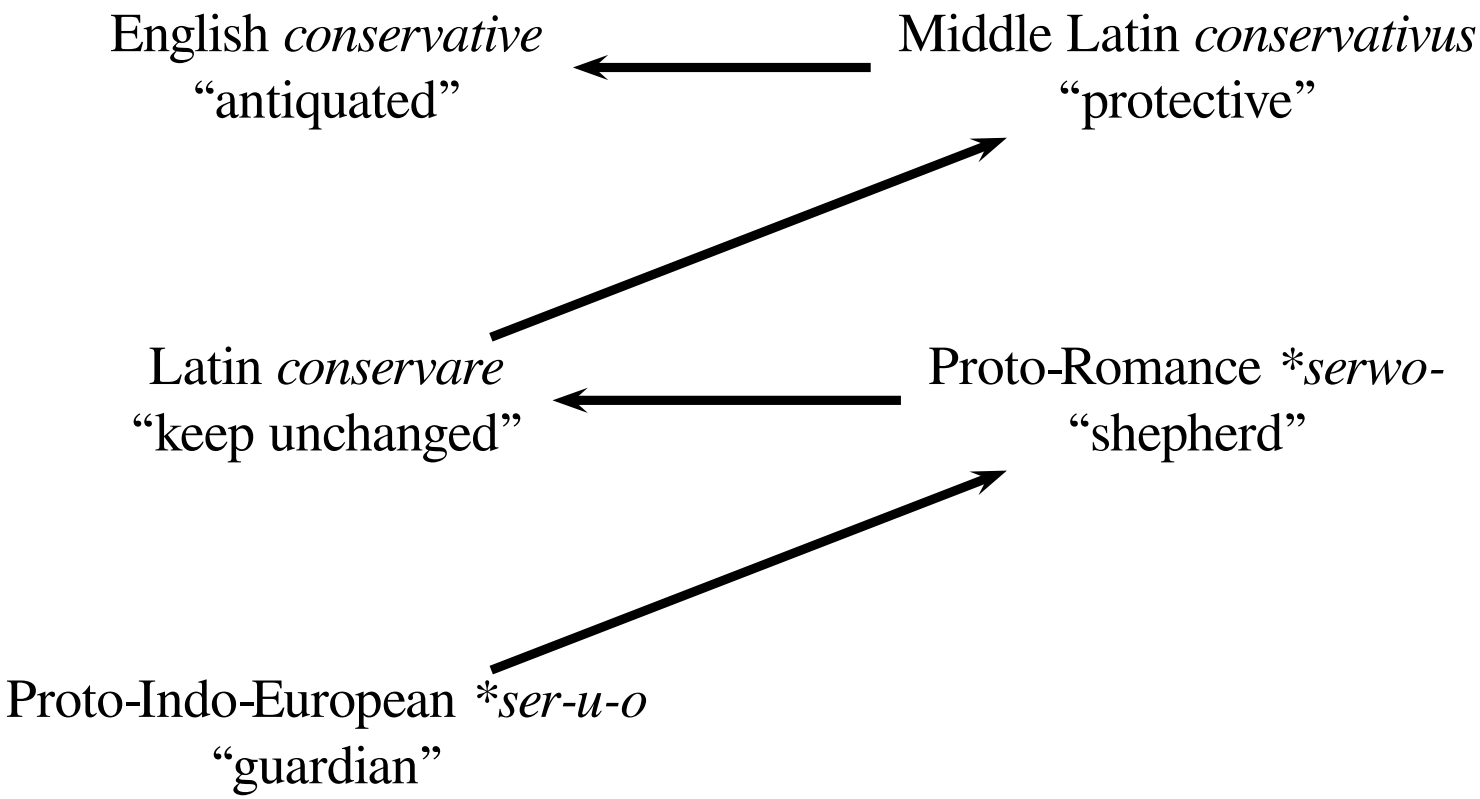
Introductory Remarks



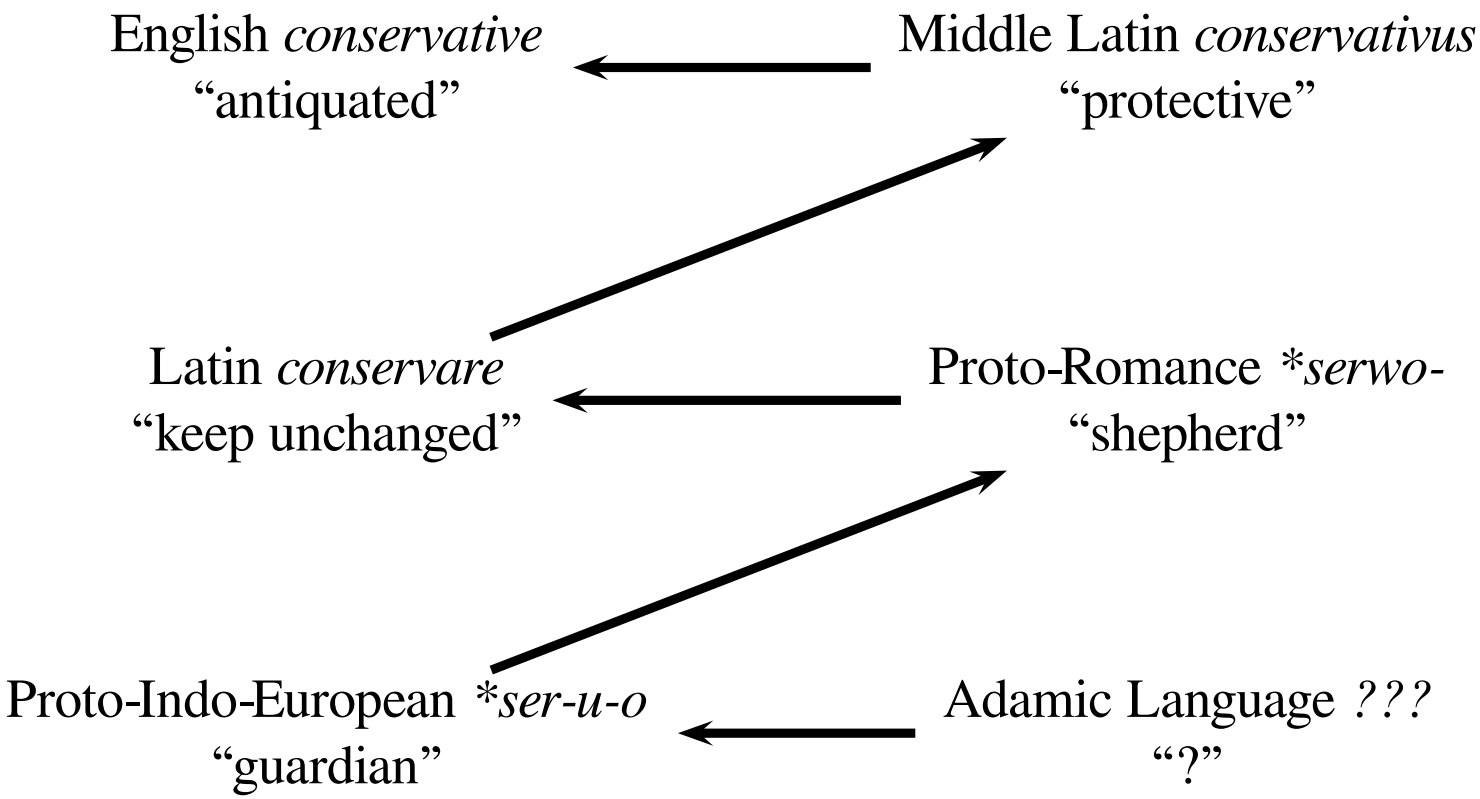
Introductory Remarks



Introductory Remarks



Introductory Remarks



Introductory Remarks

Konserve

Introductory Remarks

Konserve

Part of Speech	noun
frequency	moderate

Introductory Remarks

Konserve

Part of Speech	noun
frequency	moderate

meaning	“preserve tin”
synonyms	“Büchse”, “Konservenbüchse”, “Konservendose”
pronunciation	kɔnzɛrvə

Introductory Remarks

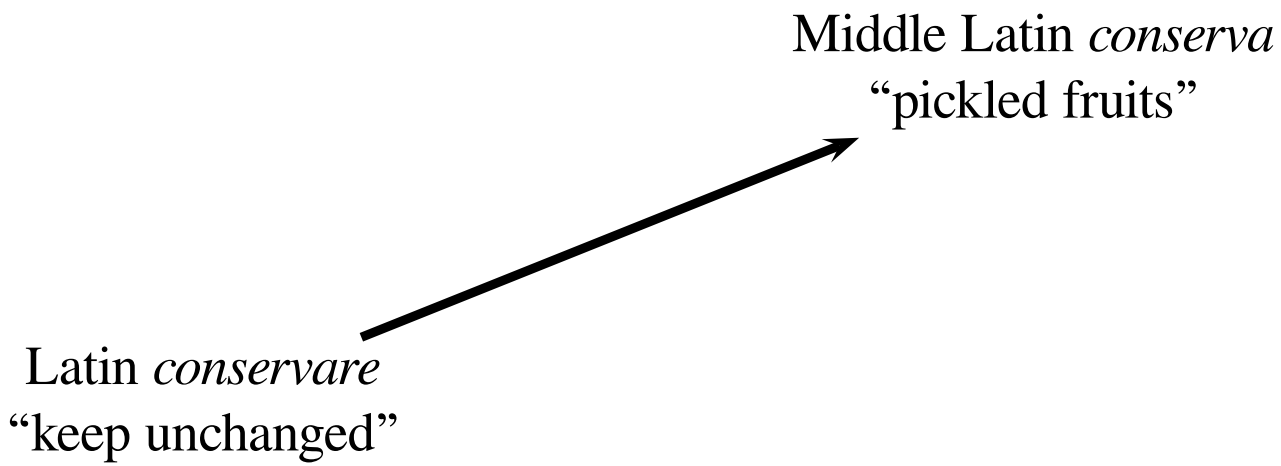
Konserve

Part of Speech	noun
frequency	moderate
meaning	“preserve tin”
synonyms	“Büchse”, “Konservenbüchse”, “Konservendose”
pronunciation	kɔnzɛrvə
origin	Middle Latin *conserva*

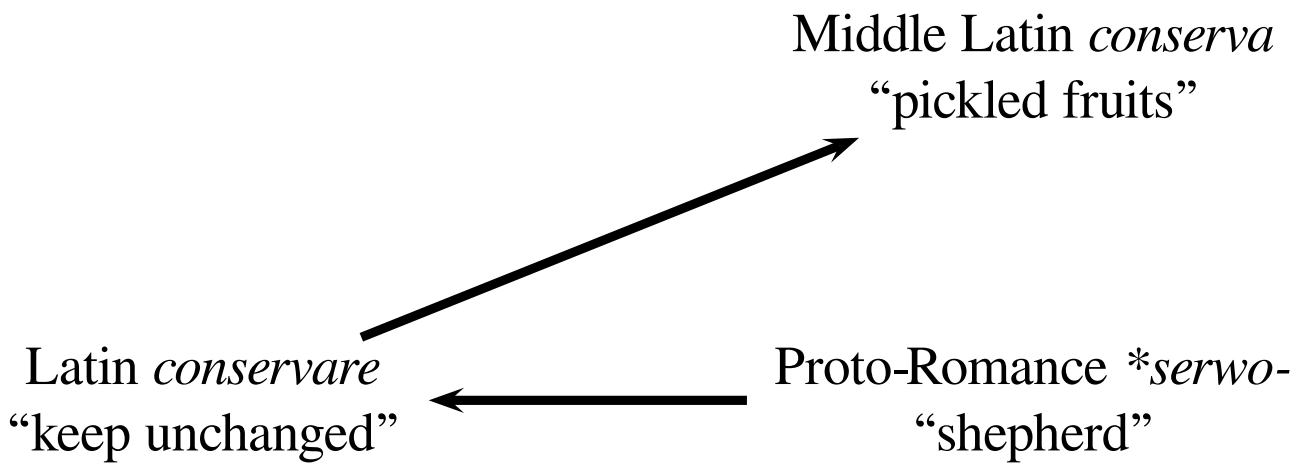
Introductory Remarks

Middle Latin *conserva*

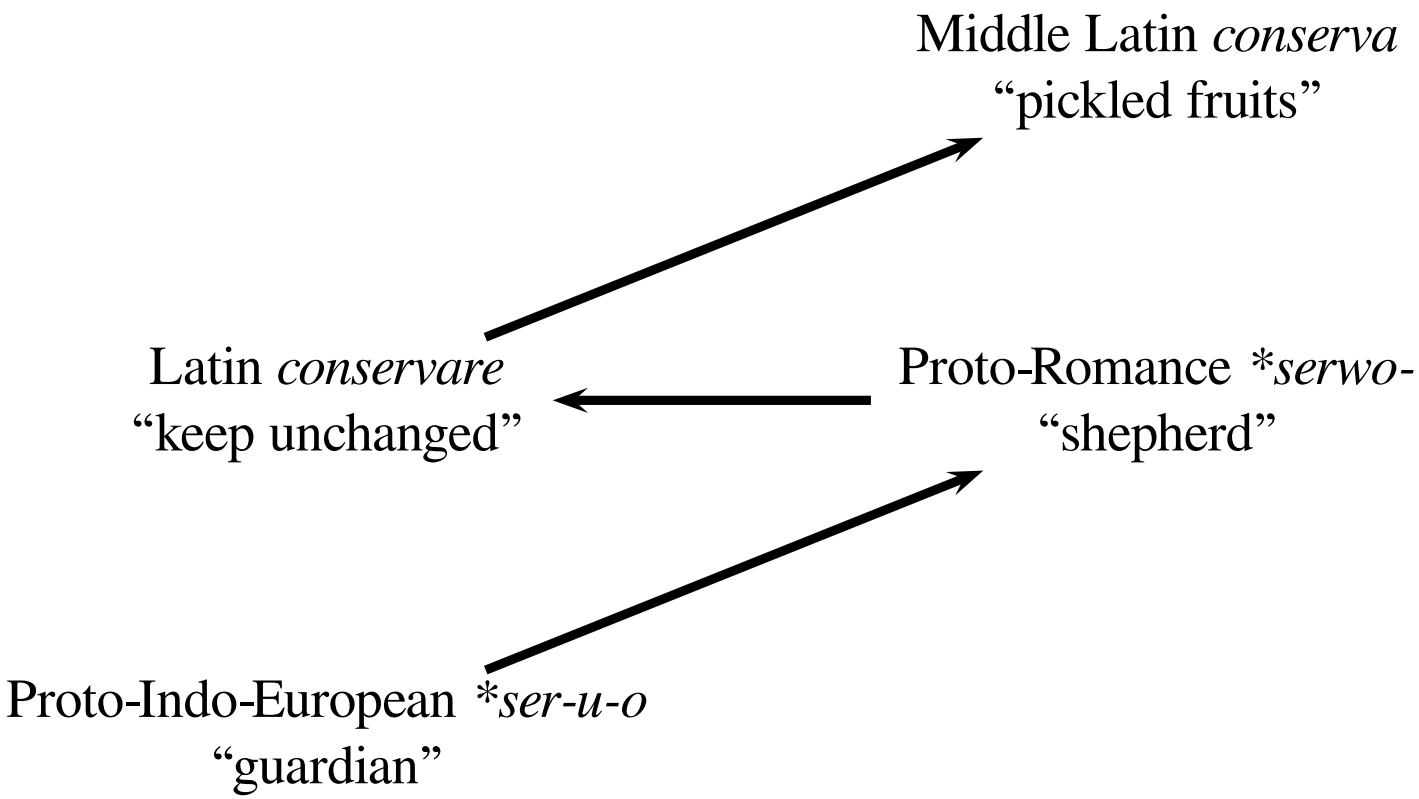
Introductory Remarks



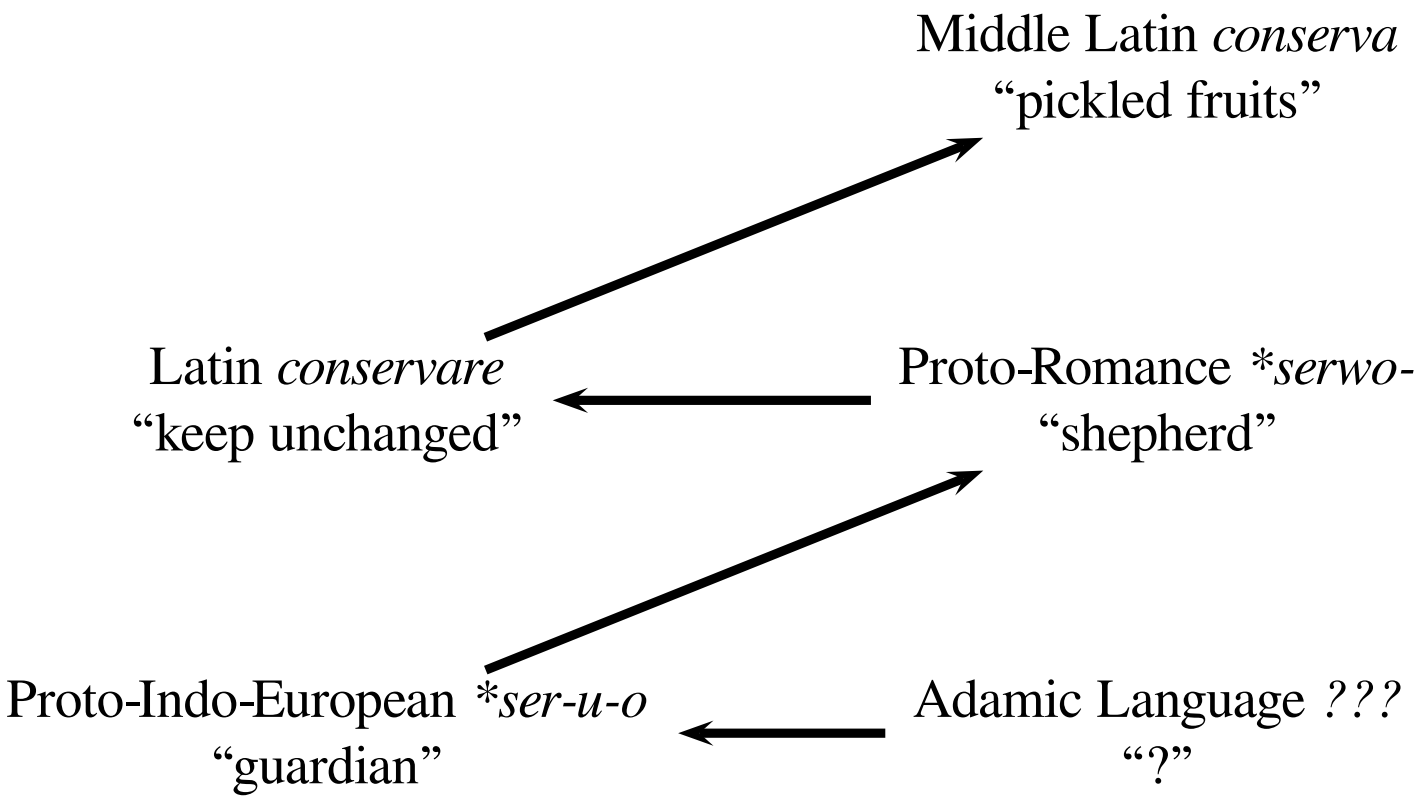
Introductory Remarks



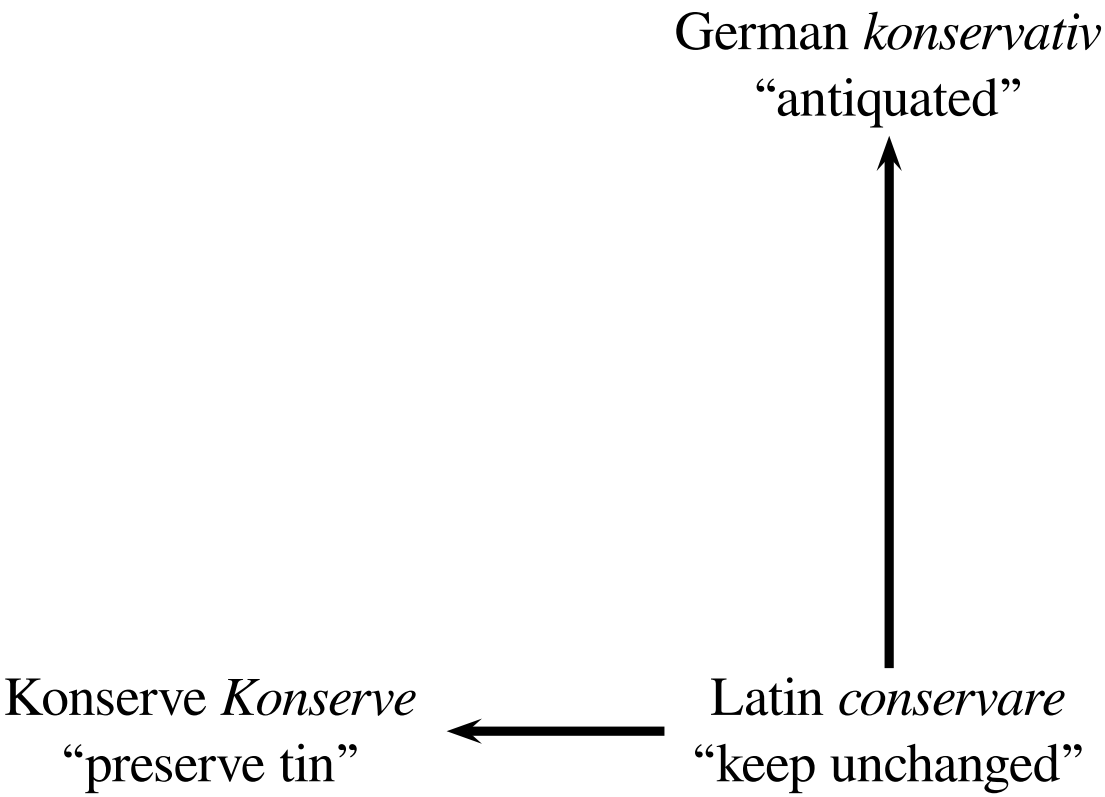
Introductory Remarks



Introductory Remarks



Introductory Remarks



Introductory Remarks

Konservativ kommt
nicht von Konserve!

Introductory Remarks

Konservativ kommt
nicht von Konserve!

Q. E. D.

Introductory Remarks

What the *** is historical linguistics?

Introductory Remarks

What the *** is historical linguistics?

Historical linguistics is a (scientific) discipline that does not specifically care about what words mean, but where they come from...

Structure of the Talk

What is LingPy?

What is LingPy?

- LingPy is a Python library (see `http://lingulist.de/lingpy`) for automatic tasks in historical linguistics.

What is LingPy?

- LingPy is a Python library (see `http://lingulist.de/lingpy`) for automatic tasks in historical linguistics.
- The current release of LingPy (lingpy-1.0) provides methods for sequence modelling, pairwise and multiple sequence alignment (SCA, List 2012a), automatic cognate detection (LexStat, List 2012b), and plotting routines (see the online documentation for details).

What is LingPy?

- LingPy is a Python library (see `http://lingulist.de/lingpy`) for automatic tasks in historical linguistics.
- The current release of LingPy (lingpy-1.0) provides methods for sequence modelling, pairwise and multiple sequence alignment (SCA, List 2012a), automatic cognate detection (LexStat, List 2012b), and plotting routines (see the online documentation for details).
- LingPy can be invoked from the Python shell or inside Python scripts (examples are given in the online documentation).

Current Features

Current Features

- tokenize phonetic sequences

Current Features

- tokenize phonetic sequences
- convert phonetic sequences into abstract sound classes

Current Features

- tokenize phonetic sequences
- convert phonetic sequences into abstract sound classes
- calculate basic prosodic characteristics of phonetic sequences

Current Features

- tokenize phonetic sequences
- convert phonetic sequences into abstract sound classes
- calculate basic prosodic characteristics of phonetic sequences
- carry out pairwise alignment analyses of phonetic sequences

Current Features

- tokenize phonetic sequences
- convert phonetic sequences into abstract sound classes
- calculate basic prosodic characteristics of phonetic sequences
- carry out pairwise alignment analyses of phonetic sequences
- carry out multiple alignment analyses of phonetic sequences

Current Features

- tokenize phonetic sequences
- convert phonetic sequences into abstract sound classes
- calculate basic prosodic characteristics of phonetic sequences
- carry out pairwise alignment analyses of phonetic sequences
- carry out multiple alignment analyses of phonetic sequences
- automatically search for cognates and etymologically related words in multilingual word lists

Basic Ideas

Basic Ideas

- **adapt common techniques for sequence comparison and phylogenetic reconstruction to the specific needs of historical linguistics (general idea)**

Basic Ideas

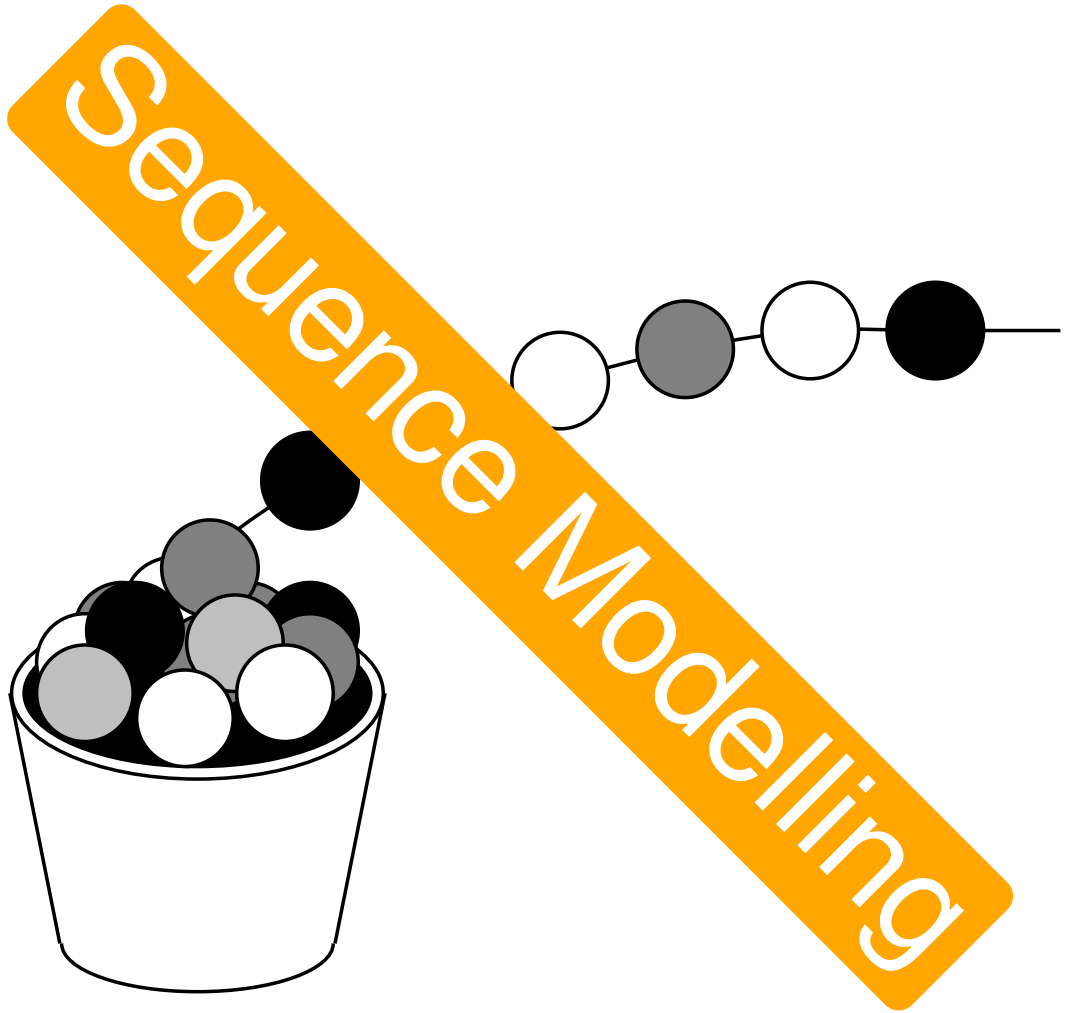
- **adapt common techniques for sequence comparison and phylogenetic reconstruction to the specific needs of historical linguistics (general idea)**
- construct realistic models of phonetic sequences by distinguishing external and internal representations (module `lingpy.sequence`)

Basic Ideas

- **adapt common techniques for sequence comparison and phylogenetic reconstruction to the specific needs of historical linguistics (general idea)**
- construct realistic models of phonetic sequences by distinguishing external and internal representations (module `lingpy.sequence`)
- compare sequences in a way that closely reflects common linguistic theory (module `lingpy.compare`)

Basic Ideas

- **adapt common techniques for sequence comparison and phylogenetic reconstruction to the specific needs of historical linguistics (general idea)**
- construct realistic models of phonetic sequences by distinguishing external and internal representations (module `lingpy.sequence`)
- compare sequences in a way that closely reflects common linguistic theory (module `lingpy.compare`)
- compare languages in a way that closely reflects the basic methods of historical linguistics (module `lingpy.lexstat`)


Sequence Modeling

Excursus: Sequence Similarity

Excursus: Sequence Similarity

Synchronic Sequence Similarity

Sequences are judged to be similar if the segments of the sequences are phonetically similar (‘phenotypic resemblence’, Lass 1997).

Excursus: Sequence Similarity

Synchronic Sequence Similarity

Sequences are judged to be similar if the segments of the sequences are phonetically similar ('phenotypic resemblence', Lass 1997).

Diachronic Sequence Similarity

Sequences are judged to be similar if the segments of the sequences correspond *systematically* ('genotypic resemblence', Lass 1997).

Excursus: Sequence Similarity

Synchronic Sequence Similarity

Diachronic Sequence Similarity

Excursus: Sequence Similarity

Synchronic Sequence Similarity

Greek	mati	'eye'	≈	Malay	mata	'eye'
Greek	θɛɔs	'god'	≈	Spanish	diɔs	'god'

Diachronic Sequence Similarity

Excursus: Sequence Similarity

Synchronic Sequence Similarity

Greek	mati	'eye'	≈	Malay	mata	'eye'
Greek	θɛɔs	'god'	≈	Spanish	diɔs	'god'

Diachronic Sequence Similarity

German	$ts^haːn$	'tooth'	≈	English	tʊːθ	'tooth'
Spanish	etʃo	'fact'	≈	French	fɛ	'fact'

Paradigmatic Aspects

Sound Classes

Paradigmatic Aspects

Sound Classes

Sounds which often occur in correspondence relations in genetically related languages can be clustered into classes (types). It is assumed “that phonetic correspondences inside a ‘type’ are more regular than those between different ‘types’” (Dolgopolsky 1986: 35).

Paradigmatic Aspects

Sound Classes

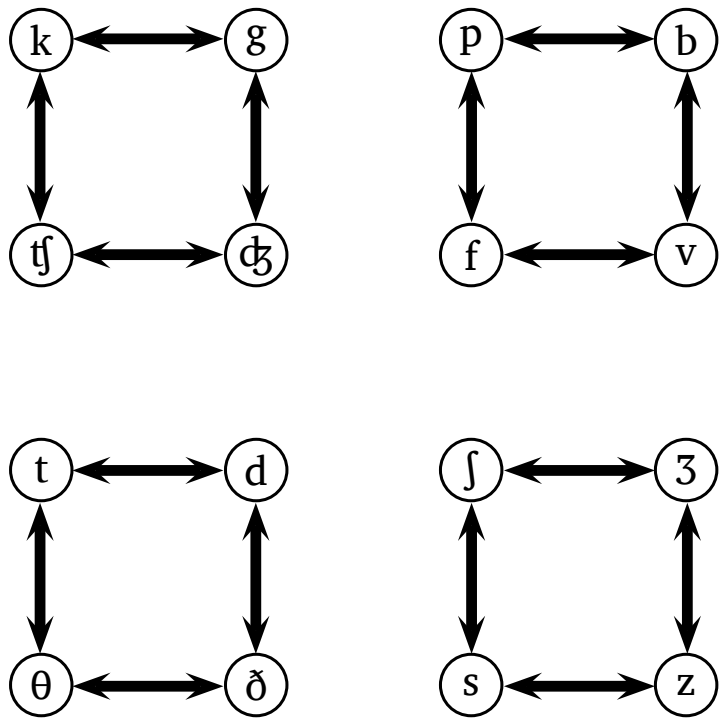
Sounds which often occur in correspondence relations in genetically related languages can be clustered into classes (types). It is assumed “that phonetic correspondences inside a ‘type’ are more regular than those between different ‘types’” (Dolgopolsky 1986: 35).

k	g	p	b
tʃ	dʒ	f	v
t	d	ʃ	ʒ
θ	ð	s	z

Paradigmatic Aspects

Sound Classes

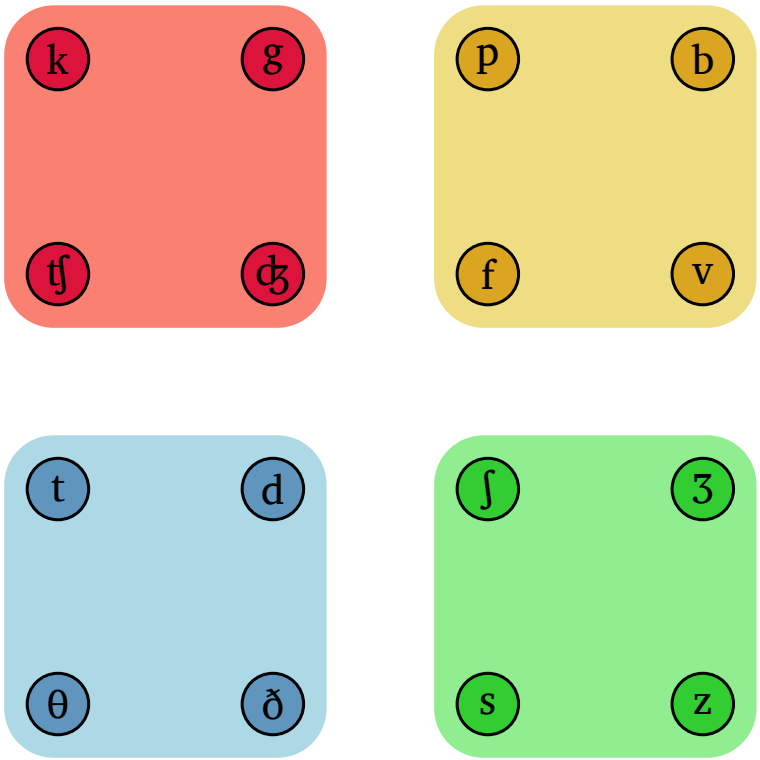
Sounds which often occur in correspondence relations in genetically related languages can be clustered into classes (types). It is assumed "that phonetic correspondences inside a 'type' are more regular than those between different 'types'" (Dolgopolsky 1986: 35).



Paradigmatic Aspects

Sound Classes

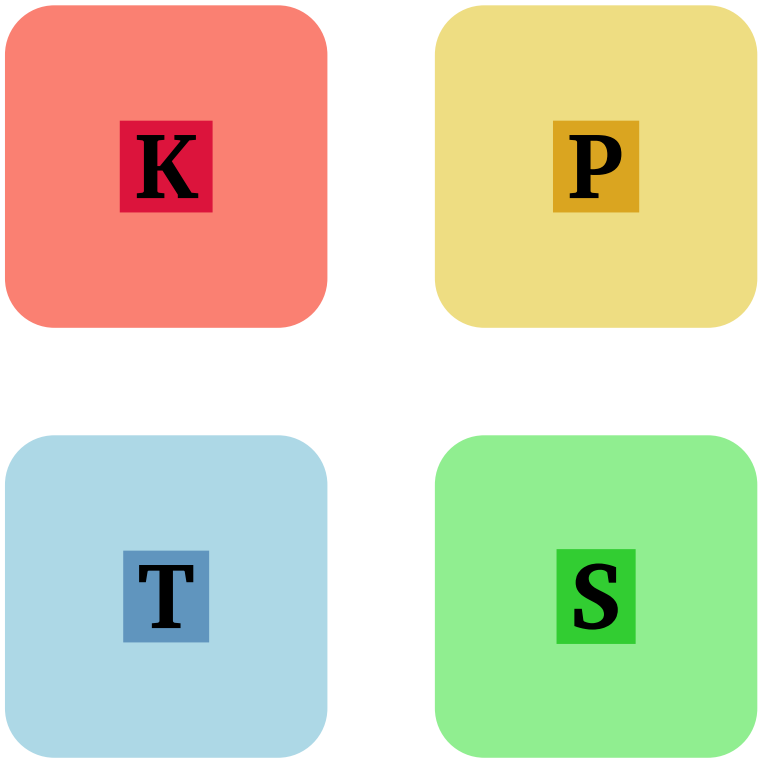
Sounds which often occur in correspondence relations in genetically related languages can be clustered into classes (types). It is assumed “that phonetic correspondences inside a ‘type’ are more regular than those between different ‘types’” (Dolgopolsky 1986: 35).



Paradigmatic Aspects

Sound Classes

Sounds which often occur in correspondence relations in genetically related languages can be clustered into classes (types). It is assumed “that phonetic correspondences inside a ‘type’ are more regular than those between different ‘types’” (Dolgopolsky 1986: 35).



Syntagmatic Aspects

Syntagmatic Aspects

- sound change occurs more frequently in prosodically **weak** positions of phonetic sequences (Geisler 1992)

Syntagmatic Aspects

- sound change occurs more frequently in prosodically **weak** positions of phonetic sequences (Geisler 1992)
- given the sonority structure of a phonetic sequence, one can distinguish positions that differ regarding their **prosodic context**

Syntagmatic Aspects

- sound change occurs more frequently in prosodically **weak** positions of phonetic sequences (Geisler 1992)
- given the sonority structure of a phonetic sequence, one can distinguish positions that differ regarding their **prosodic context**
- prosodic context can be modelled by representing a sequence by a **prosodic string**, indicating the different prosodic contexts of each segment

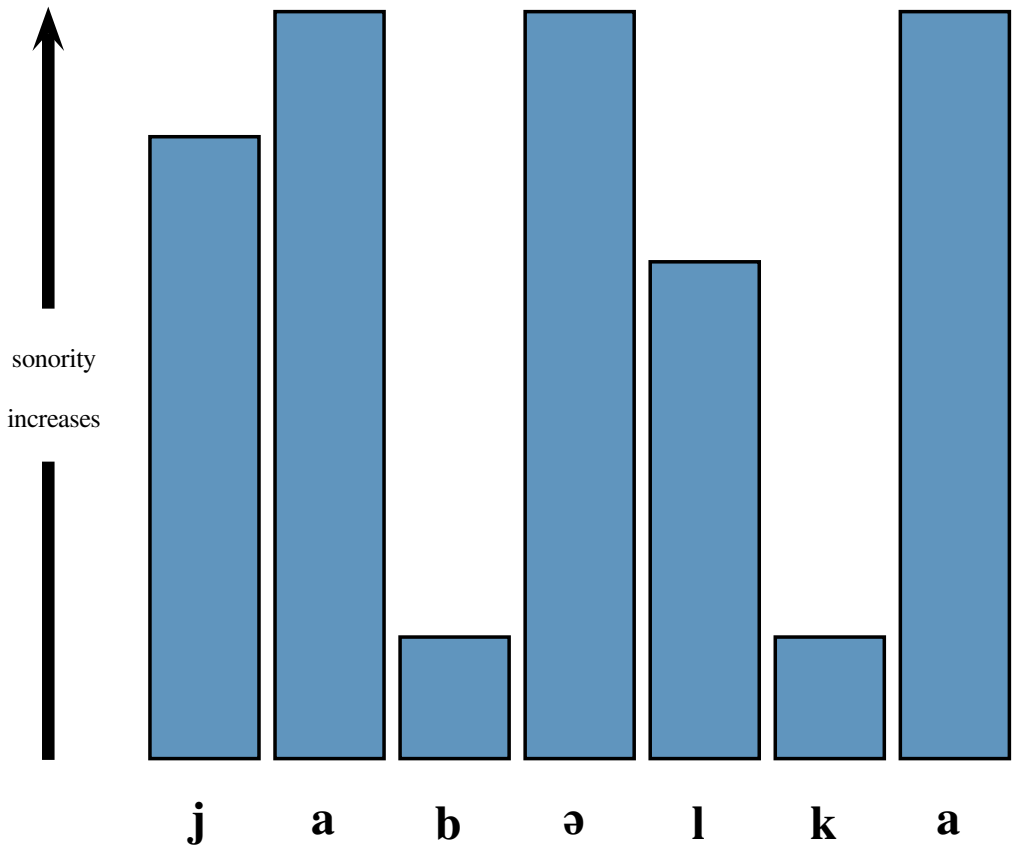
Syntagmatic Aspects

- sound change occurs more frequently in prosodically **weak** positions of phonetic sequences (Geisler 1992)
- given the sonority structure of a phonetic sequence, one can distinguish positions that differ regarding their **prosodic context**
- prosodic context can be modelled by representing a sequence by a **prosodic string**, indicating the different prosodic contexts of each segment
- based on the relative strength of all sites in a phonetic sequence, substitution scores and gap penalties can be modified when carrying out alignment analyses

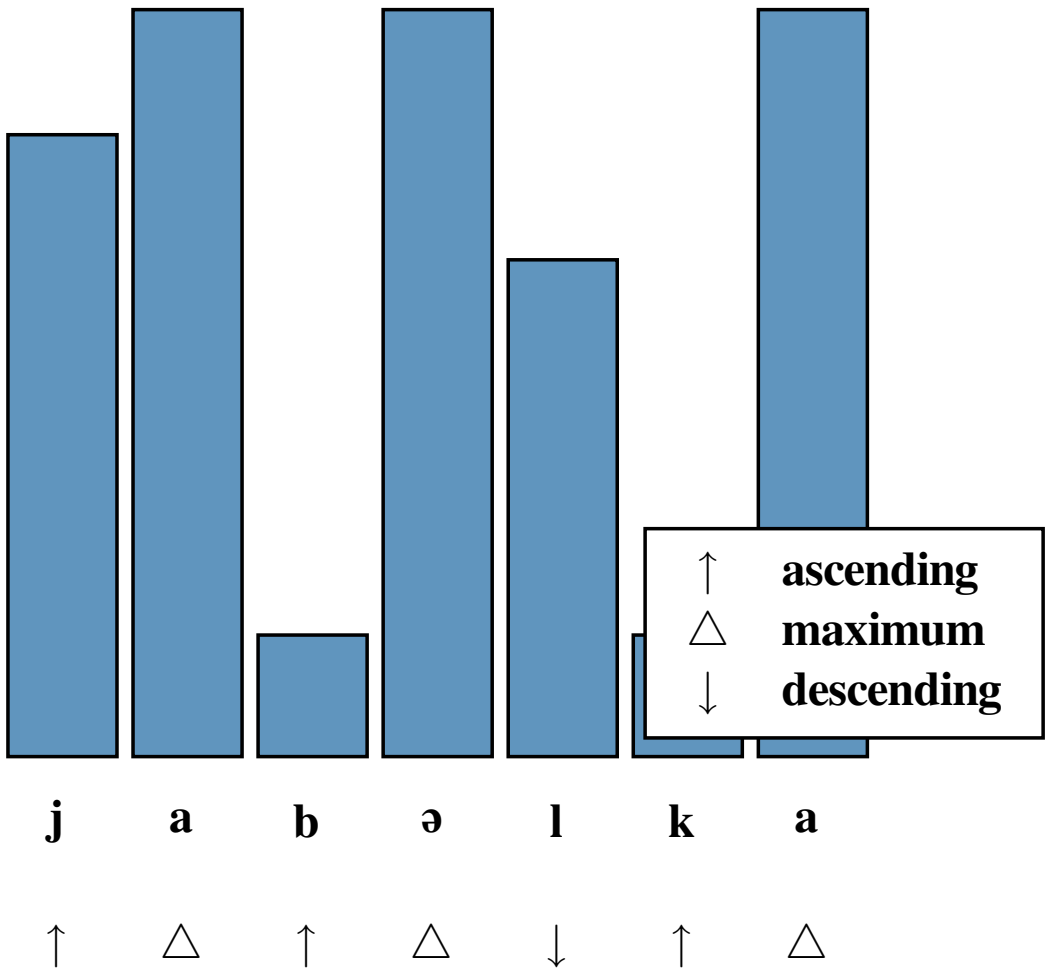
Syntagmatic Aspects

j a b ə l k a

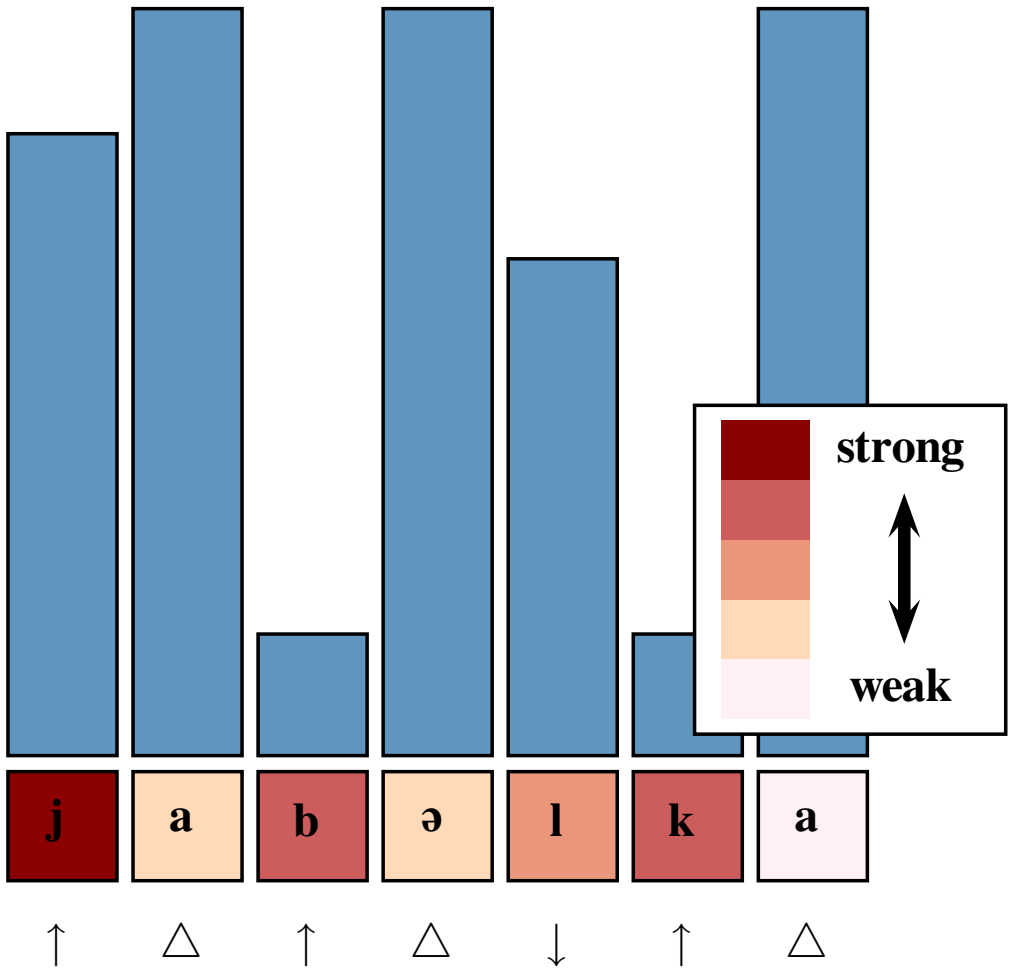
Syntagmatic Aspects



Syntagmatic Aspects



Syntagmatic Aspects



Examples

Examples

```
>>> from lingpy import *
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvati:f")
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvati:f")
>>> print ' '.join(konservativ.tokens)
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvati:f")
>>> print ' '.join(konservativ.tokens)
k ɔ n z ɛ r v a t i: f
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvati:f")
>>> print ' '.join(konservativ.tokens)
k ɔ n z ɛ r v a t i: f
>>> print konservativ.classes
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvatiːf")
>>> print ' '.join(konservativ.tokens)
k ɔ n z ɛ r v a t iː f
>>> print konservativ.classes
KUNSERBATIB
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvati:f")
>>> print ' '.join(konservativ.tokens)
k ɔ n z ɛ r v a t i: f
>>> print konservativ.classes
KUNSERBATIB
>>> print ' '.join([str(i) for i in konservativ.sonar])
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvati:f")
>>> print ' '.join(konservativ.tokens)
k ɔ n z ɛ r v a t i: f
>>> print konservativ.classes
KUNSERBATIB
>>> print ' '.join([str(i) for i in konservativ.sonar])
1 7 4 3 7 5 3 7 1 7 3
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvati:f")
>>> print ' '.join(konservativ.tokens)
k ɔ n z ɛ r v a t i: f
>>> print konservativ.classes
KUNSERBATIB
>>> print ' '.join([str(i) for i in konservativ.sonar])
1 7 4 3 7 5 3 7 1 7 3
>>> print konservativ.prostring
```

Examples

```
>>> from lingpy import *
>>> konservativ = Sequence("kɔnzɛrvatiːf")
>>> print ' '.join(konservativ.tokens)
k ɔ n z ɛ r v a t iː f
>>> print konservativ.classes
KUNSERBATIB
>>> print ' '.join([str(i) for i in konservativ.sonar])
1 7 4 3 7 5 3 7 1 7 3
>>> print konservativ.prostring
#VcCvcCvCv$
```

Phonetic Alignment

Excursus: Alignment Analyses

Alignment Analyses

Excursus: Alignment Analyses

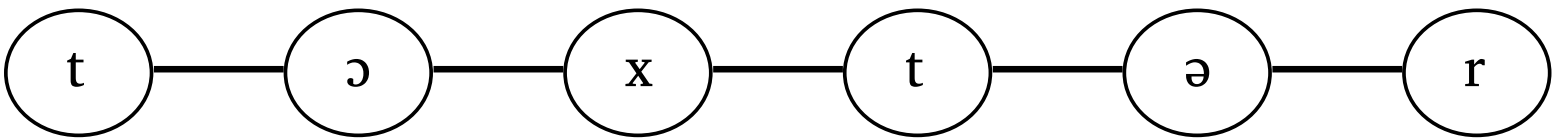
Alignment Analyses

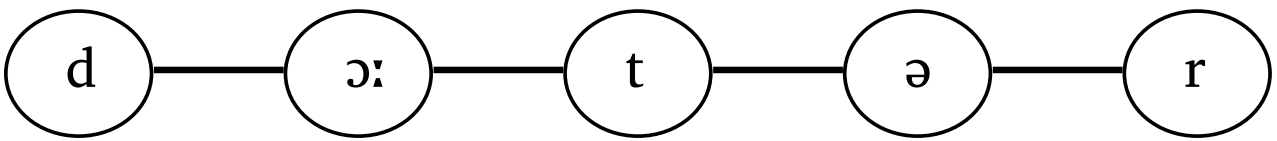
In alignment analyses, sequences are arranged in a matrix in such a way that corresponding elements occur in the same column, while empty cells resulting from non-corresponding elements are filled with gap symbols.

Excursus: Alignment Analyses

Alignment Analyses

In alignment analyses, sequences are arranged in a matrix in such a way that corresponding elements occur in the same column, while empty cells resulting from non-corresponding elements are filled with gap symbols.

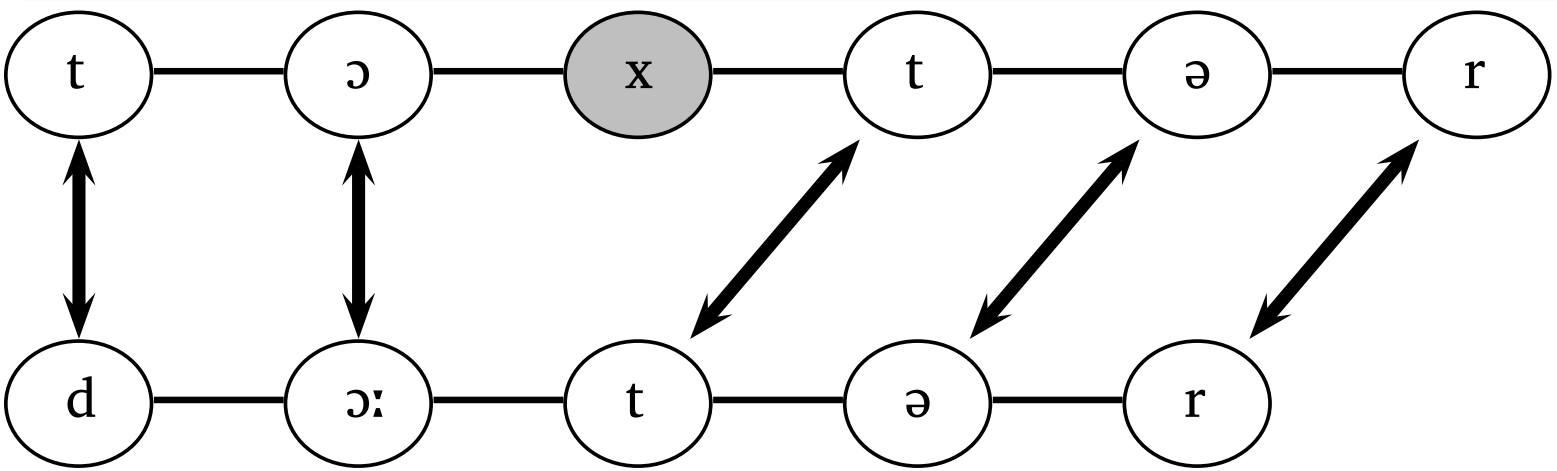




Excursus: Alignment Analyses

Alignment Analyses

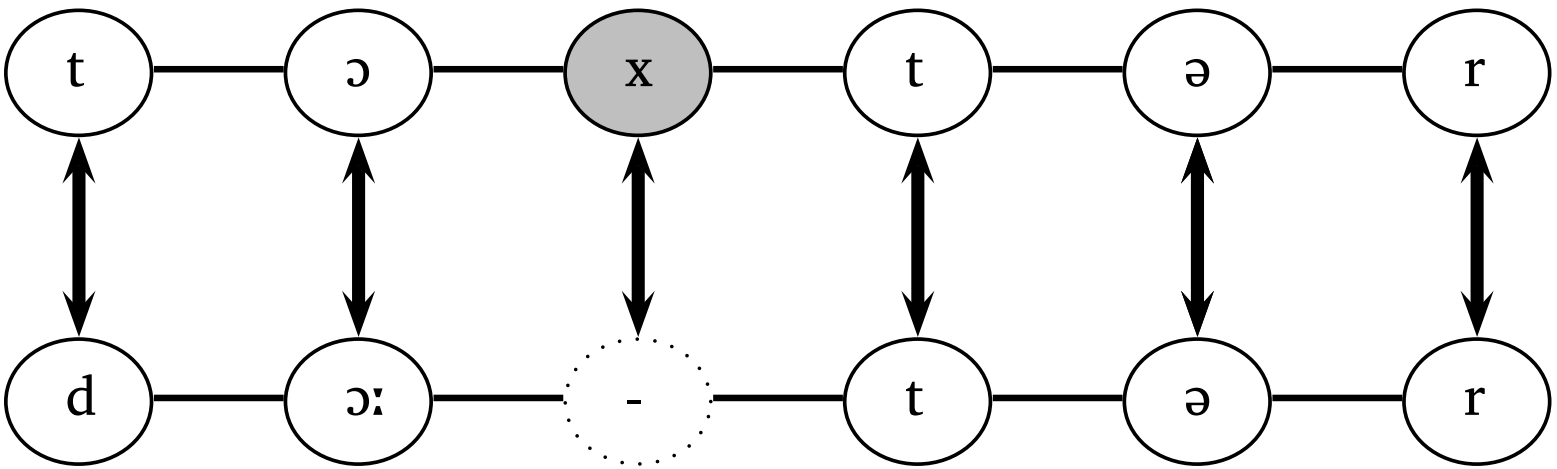
In alignment analyses, sequences are arranged in a matrix in such a way that corresponding elements occur in the same column, while empty cells resulting from non-corresponding elements are filled with gap symbols.



Excursus: Alignment Analyses

Alignment Analyses

In alignment analyses, sequences are arranged in a matrix in such a way that corresponding elements occur in the same column, while empty cells resulting from non-corresponding elements are filled with gap symbols.



Phonetic Alignment

Sound-Class Based Alignment (SCA, List 2012a)

Phonetic Alignment

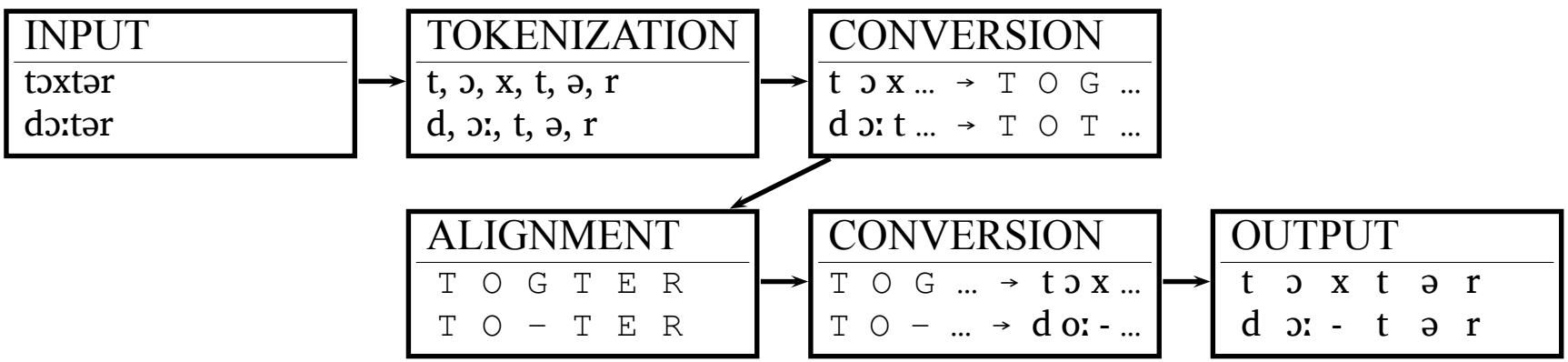
Sound-Class Based Alignment (SCA, List 2012a)

Sound classes and alignment analyses can be easily combined by representing phonetic sequences internally as sound classes and comparing the sound classes with traditional alignment algorithms.

Phonetic Alignment

Sound-Class Based Alignment (SCA, List 2012a)

Sound classes and alignment analyses can be easily combined by representing phonetic sequences internally as sound classes and comparing the sound classes with traditional alignment algorithms.



Example

```
>>> from lingpy import *
```

Example

```
>>> from lingpy import *
>>> konservativ = "kɔnzɛrvati:f"
```

Example

```
>>> from lingpy import *
>>> konservativ = "kɔnzɛrvati:f"
>>> Konserve = "kɔnzɛrvə"
```

Example

```
>>> from lingpy import *
>>> konservativ = "kɔnzɛrvati:f"
>>> Konserve = "kɔnzɛrvə"
>>> pair = Pairwise(konservativ,Konserve)
```

Example

```
>>> from lingpy import *
>>> konservativ = "kɔnzɛrvati:f"
>>> Konserve = "kɔnzɛrvə"
>>> pair = Pairwise(konservativ,Konserve)
>>> pair.align()
```

Example

```
>>> from lingpy import *
>>> konservativ = "kɔnzɛrvati:f"
>>> Konserve = "kɔnzɛrvə"
>>> pair = Pairwise(konservativ,Konserve)
>>> pair.align()
>>> pair.distance()
```

Example

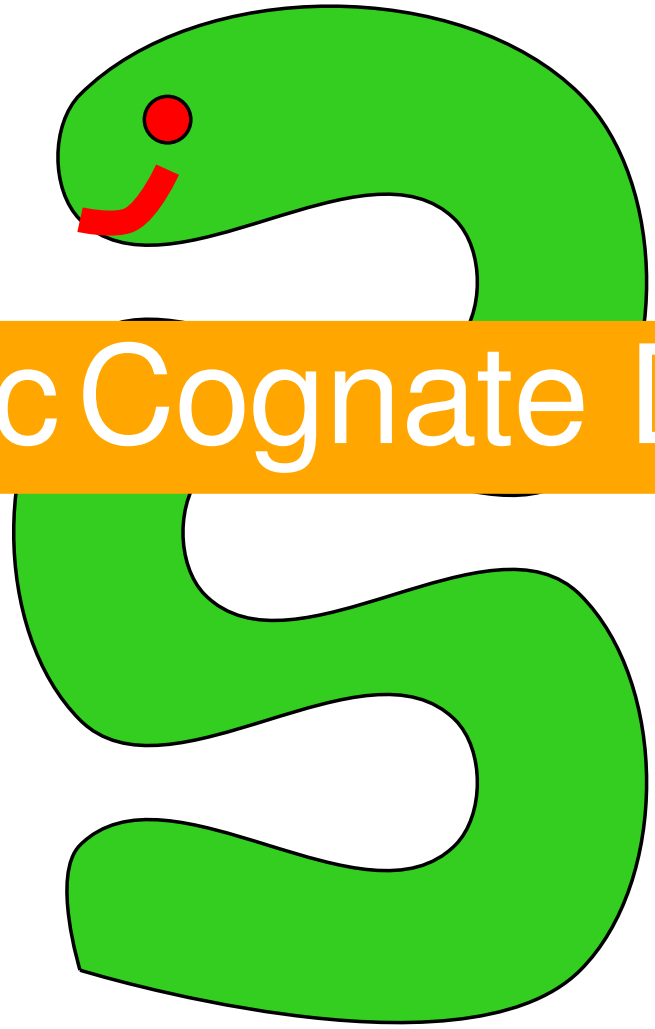
```
>>> from lingpy import *
>>> konservativ = "kɔnzɛrvati:f"
>>> Konserve = "kɔnzɛrvə"
>>> pair = Pairwise(konservativ,Konserve)
>>> pair.align()
>>> pair.distance()
>>> print pair
```

Example

```
>>> from lingpy import *
>>> konservativ = "kɔnzɛrvatiːf"
>>> Konserve = "kɔnzɛrvə"
>>> pair = Pairwise(konservativ,Konserve)
>>> pair.align()
>>> pair.distance()
>>> print pair
k  ɔ  n  z  ɛ  r  v  a  t  iː f
k  ɔ  n  z  ɛ  r  v  -  -  ə  -
0.3
```

Example

Taxon	Alignment									
Xiangtan	-	i	-	24	d	əɯ	12	-	-	-
Qingdao	-	i	-	42	tʰ	ou	-	-	-	-
Xi_an	-	ə	r	21	tʰ	ou	-	-	-	-
Wuhan	-	ɯ	-	213	tʰ	əu	-	-	-	-
Guangzhou	j	i	t	2	tʰ	ɐu	21	-	-	-
Nanning	j	i	t	22	tʰ	ɐu	21	-	-	-
Xianggang	j	ɐ	t	2	tʰ	ɐu	21	-	-	-
Nanning	j	ɐ	t	22	tʰ	ɐu	21	-	-	-
Taibei	l	i	t	44	tʰ	au	24	-	-	-
Xiamen	l	i	t	5	tʰ	au	-	-	-	-
Jian_ou	m	i	-	33	tʰ	e	33	-	-	-
Shexian	n	i	-	22	tʰ	iu	44	-	-	-
Fuzhou	n	i	ʔ	5	tʰ	au	53	-	-	-
Shantou	z	i	k	5	-	-	-	-	-	-
Shantou	z	i	k	5	tʰ	au	55	-	-	-
Haikou	z	i	t	3	-	-	-	-	-	-
Haikou	z	i	t	3	h	au	31	-	-	-
Taoyuan	ŋ	i	t	22	tʰ	eu	11	-	-	-
Meixian	ŋ	i	t	1	tʰ	eu	11	-	-	-
Wenzhou	ɳ	i	-	213	d	ɤu	-	-	-	-
Wenzhou	ɳ	i	-	213	d	ɤu	31	v	ai	213
Nanchang	ɳ	i	ʔ	5	tʰ	u	02	-	-	-
Tunxi	ɳ	ie	-	11	tʰ	iu	44	-	-	-
Suzhou	ɳ	iə	ʔ	3	d	ʏ	13	-	-	-
Shanghai	ɳ	ɪ	ʔ	1	d	ɤ	13	-	-	-
Kunming	ʐ	ə	-	31	tʰ	əu	31	-	-	-
Hefei	ʐ	ə	ʔ	5	tʰ	ɯ	-	-	-	-
Xining	ʐ	ɛ	-	44	tʰ	ɯ	24	-	ɛ	24
Jinan	ʐ	ɿ	-	21	tʰ	ou	-	-	-	-
Changsha	ʐ	ɿ	-	24	t	əu	-	-	-	-
Zhengzhou	ʐ	ɿ	-	24	tʰ	ou	-	-	-	-
Lanzhou	ʐ	ɿ	-	13	tʰ	ou	13	-	-	-
Yinchuan	ʐ	ɿ	-	13	tʰ	əu	-	-	-	-
Haerbin	ʐ	ɿ	-	53	tʰ	ou	-	-	-	-
Beijing	ʐ	ɿ	-	51	tʰ	ou	1	-	-	-
Huhehaote	ʐ	ɿ	-	55	tʰ	əu	31	-	-	-
Pingyao	ʐ	ʌ	ʔ	53	t	əu	13	-	iᴇ	13



Automatic Cognate Detection

Excursus: Language-Specific Similarities

Language-Specific Similarity Measure

Excursus: Language-Specific Similarities

Language-Specific Similarity Measure

- recall the aforementioned distinction between **synchronic** and **diachronic** similarity

Excursus: Language-Specific Similarities

Language-Specific Similarity Measure

- recall the aforementioned distinction between **synchronic** and **diachronic** similarity
- **diachronic** similarity is the basic of all traditional approaches to cognate detection

Excursus: Language-Specific Similarities

Language-Specific Similarity Measure

- recall the aforementioned distinction between **synchronic** and **diachronic** similarity
- **diachronic** similarity is the basic of all traditional approaches to cognate detection
- diachronic sequence similarity is determined on the basis of systematic sound correspondences as opposed to similarity based on surface resemblances of phonetic segments

Excursus: Language-Specific Similarities

Language-Specific Similarity Measure

- recall the aforementioned distinction between **synchronic** and **diachronic** similarity
- **diachronic** similarity is the basic of all traditional approaches to cognate detection
- diachronic sequence similarity is determined on the basis of systematic sound correspondences as opposed to similarity based on surface resemblances of phonetic segments
- the most crucial aspect of correspondence-based (diachronic) similarity is that it is **language-specific**

Excursus: Language-Specific Similarities

Language-Specific Similarity Measure

- recall the aforementioned distinction between **synchronic** and **diachronic** similarity
- **diachronic** similarity is the basic of all traditional approaches to cognate detection
- diachronic sequence similarity is determined on the basis of systematic sound correspondences as opposed to similarity based on surface resemblances of phonetic segments
- the most crucial aspect of correspondence-based (diachronic) similarity is that it is **language-specific**

Meaning	German	Dutch	English
"tooth"	*Zahn* [ts aːn]	*tand* [t ɑnt]	*tooth* [t oːθ]
"ten"	*zehn* [ts eːn]	*tien* [t iːn]	*ten* [t ɛn]
"tongue"	*Zunge* [ts ʊŋə]	*tong* [t ɔŋ]	*tongue* [t ʌŋ]

Excursus: Language-Specific Similarities

Language-Specific Similarity Measure

- recall the aforementioned distinction between **synchronic** and **diachronic** similarity
- **diachronic** similarity is the basic of all traditional approaches to cognate detection
- diachronic sequence similarity is determined on the basis of systematic sound correspondences as opposed to similarity based on surface resemblances of phonetic segments
- the most crucial aspect of correspondence-based (diachronic) similarity is that it is **language-specific**

Meaning	Shanghai	Beijing	Guangzhou
“nine”	[tɕ iɤ^{35}]	Beijing [tɕ iou^{214}]	[k ɐu^{35}]
“today”	[tɕ iŋ^{55}tsɔ^{21}]	Beijing [tɕ iɚ^{55}]	[k ɐm^{53}jɐt^{2}]
“rooster”	[koŋ^{55} tɕ i^{21}]	Beijing[kuŋ^{55} tɕ i^{55}]	[k ɐi^{55}koŋ^{55}]

Automatic Cognate Detection

LexStat (List 2012b)

Automatic Cognate Detection

LexStat (List 2012b)

- LexStat is a method for the automatic detection of cognates in multilingual word lists

Automatic Cognate Detection

LexStat (List 2012b)

- LexStat is a method for the automatic detection of cognates in multilingual word lists
- LexStat based its cognate judgments on language-specific, diachronic similarity

Automatic Cognate Detection

LexStat (List 2012b)

- LexStat is a method for the automatic detection of cognates in multilingual word lists
- LexStat based its cognate judgments on language-specific, diachronic similarity
- LexStat combines the most recent approaches to sequence comparison in computer science and biology with a novel approach to sequence modelling

Automatic Cognate Detection

LexStat (List 2012b)

- LexStat is a method for the automatic detection of cognates in multilingual word lists
- LexStat based its cognate judgments on language-specific, diachronic similarity
- LexStat combines the most recent approaches to sequence comparison in computer science and biology with a novel approach to sequence modelling
- LexStat largely outperforms alternative approaches and yields 90% precision and 84% recall on very divergent language families, such as Indo-European and Austronesian

Automatic Cognate Detection

LexStat (List 2012b)

- LexStat is a method for the automatic detection of cognates in multilingual word lists
- LexStat based its cognate judgments on language-specific, diachronic similarity
- LexStat combines the most recent approaches to sequence comparison in computer science and biology with a novel approach to sequence modelling
- LexStat largely outperforms alternative approaches and yields 90% precision and 84% recall on very divergent language families, such as Indo-European and Austronesian
- LexStat yields **transparent** decisions which can be directly examined by the researcher

Automatic Cognate Detection

ID	Items	German	English	Swedish
1	hand	hant	hænd	hand
2	woman	fraʊ	wʊmən	kvina
3	know	kɛnən	nəʊ	çɛna
3	know	vɪsən	-	ve:ta
…	…	…	…	…

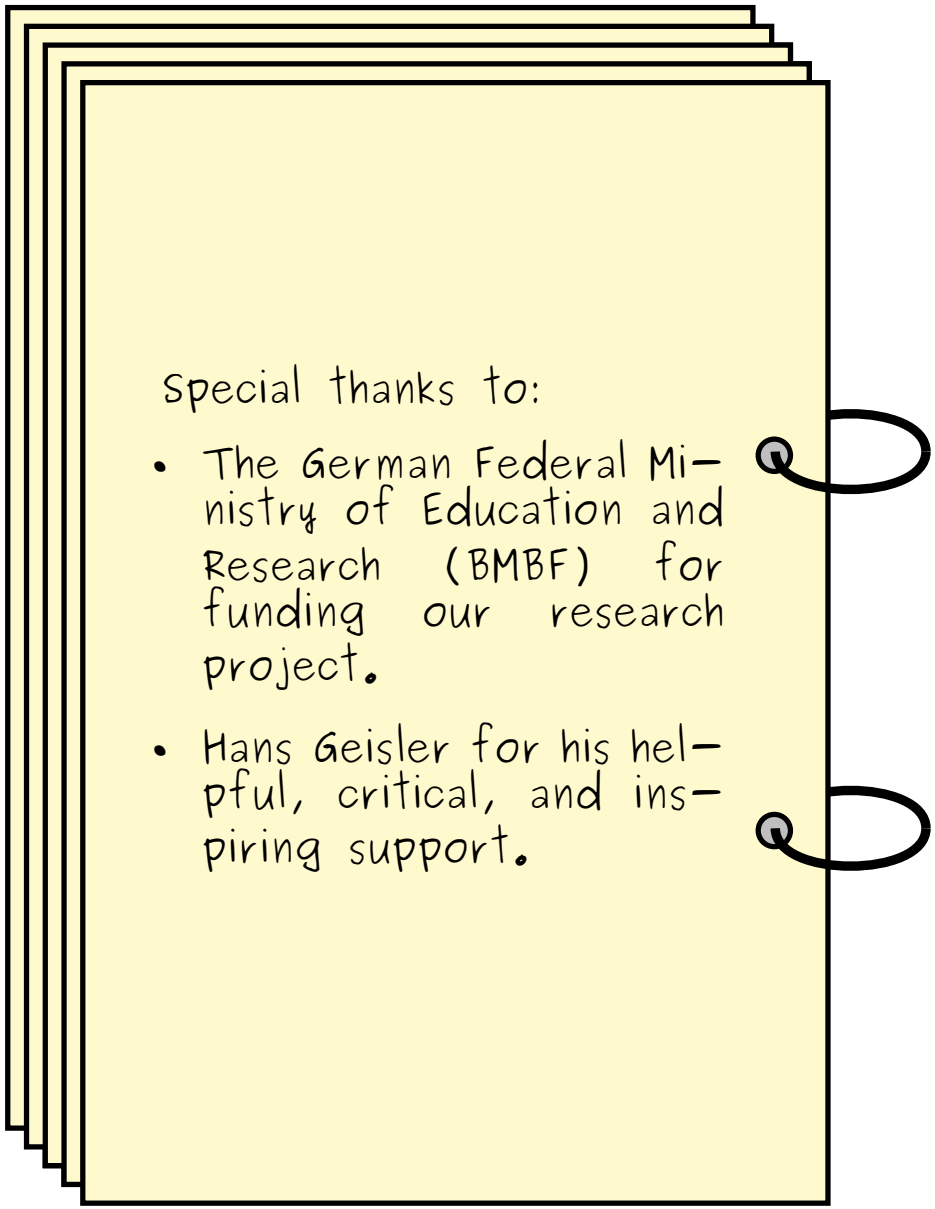
Automatic Cognate Detection

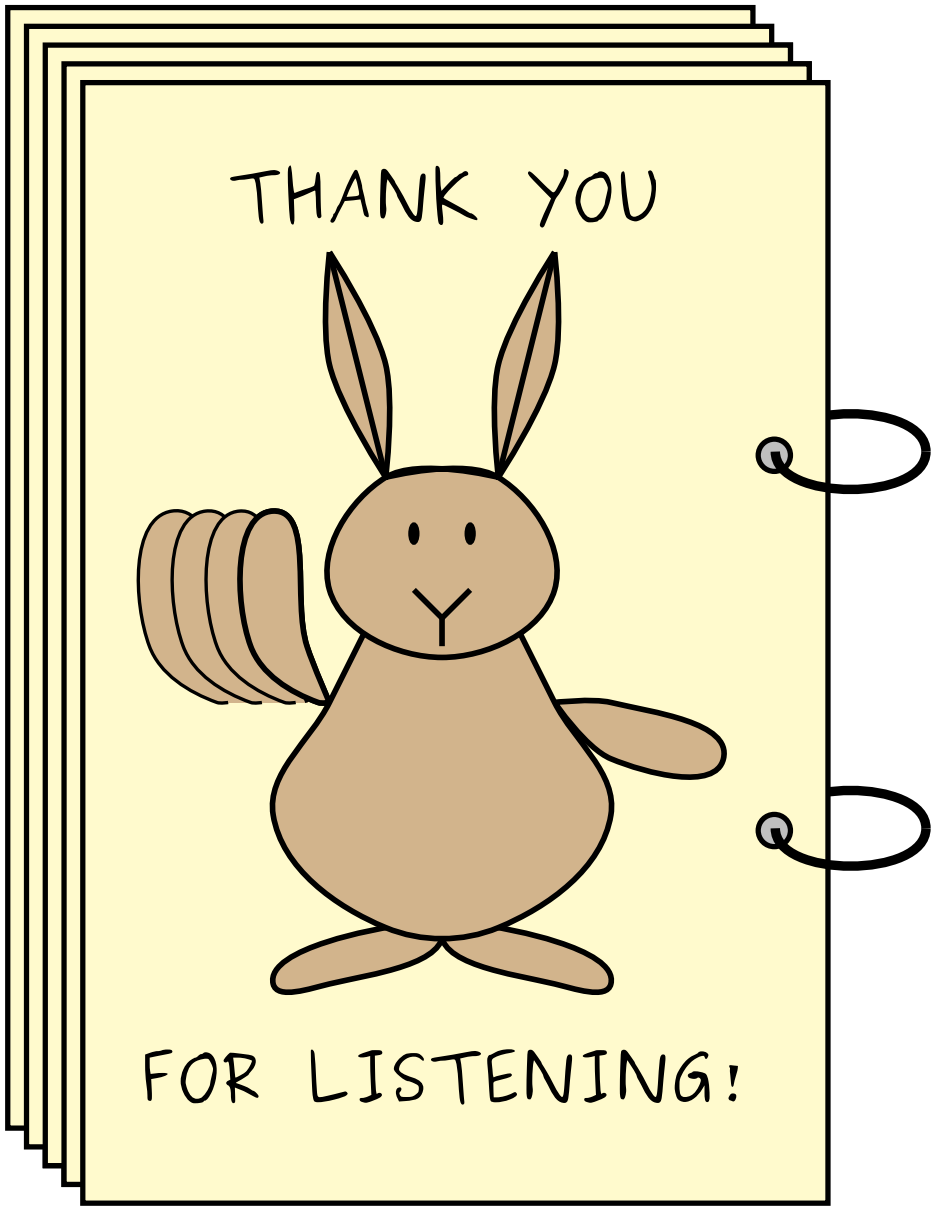
ID	Items	German	COG	English	COG	Swedish	COG
1	hand	hant	1	hænd	1	hand	1
2	woman	fraʊ	2	wʊmən	3	kvina	4
3	know	kɛnən	5	nəʊ	5	çɛna	5
3	know	vɪsən	6	-	0	ve:ta	6
…	…	…	…	…	…	…	…

Automatic Cognate Detection

Basic Concept: *belly* (ID: 4)			
CogID	**Language**	**Entry**	**Aligned Entry**
6	Danish	ɔnʌliwˀ	--
7	German	baux	b au x
7	Dutch	bœyk	b œy k
7	Swedish	buk	b u k
7	Norwegian	bʉːk	b ʉː k
8	English	bɛlɪ	--
9	Swedish	maːge	m aː g e
9	Norwegian	mɑːgə	m ɑː g ə
9	Danish	mæːvə	m æː v ə
10	Icelandic	kʰvɪːðʏr	--


What's next?
*deh3-


Special thanks to:
• The German Federal Ministry of Education and Research (BMBF) for funding our research project.
• Hans Geisler for his helpful, critical, and inspiring support.


THANK YOU
FOR LISTENING!