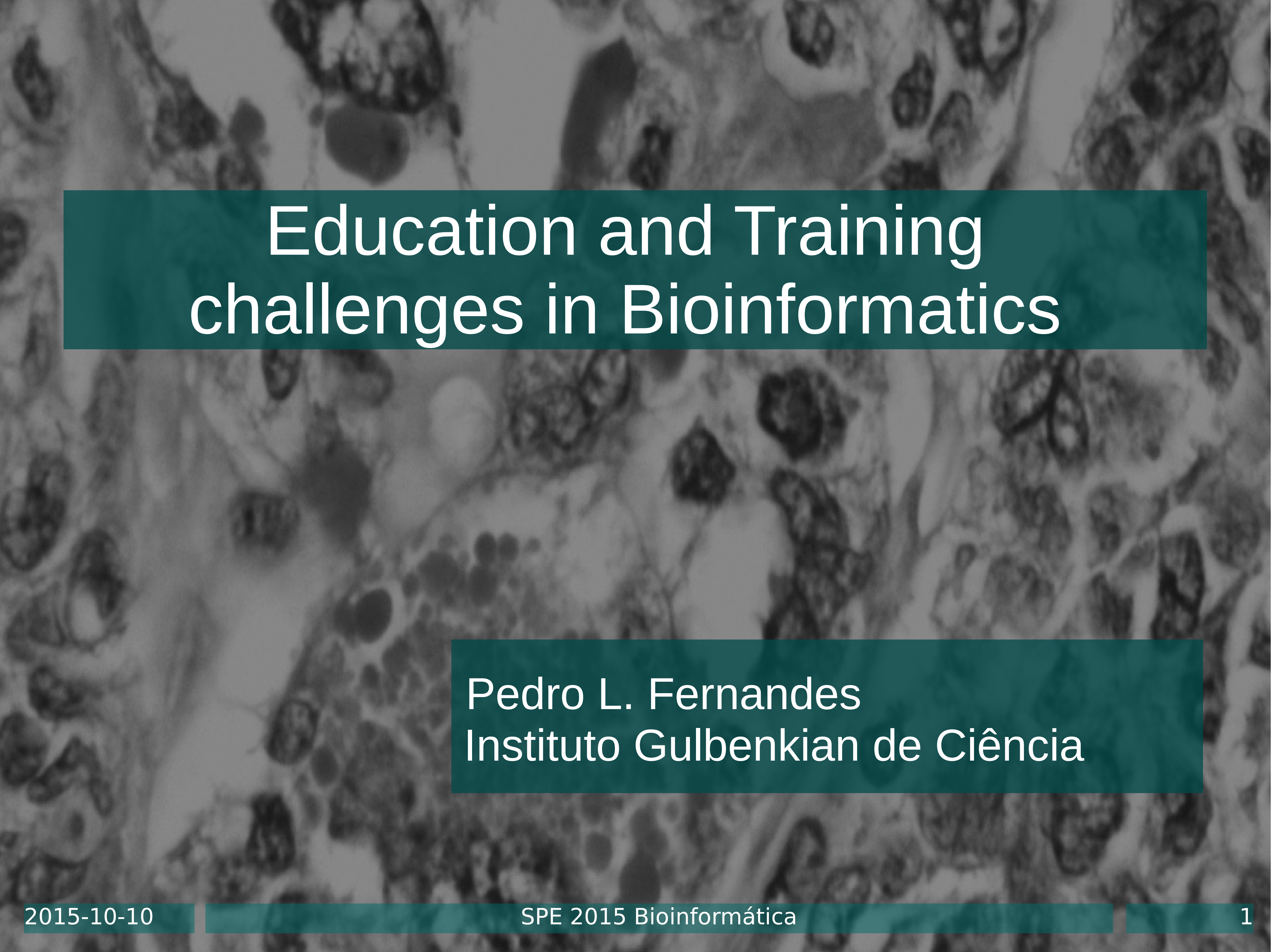


Education and Training challenges in Bioinformatics

Pedro L. Fernandes
Instituto Gulbenkian de Ciência

Why Bioinformatics?

Bioinformatics is a hybrid discipline that has a quite simple purpose:

To contribute to better understand Biology by using

biological information

and

computational methods

Biological Information

Biological information is a term that encapsulates a very wide range of entities, emanating from laboratory instruments, field observations, clinical activity, etc.

Biological information is ideally organised into databases. Datasets can be quite large and full of **variation** and **noise**.

There is a huge field of application for computer science and statistics that results from the need to manipulate such datasets.

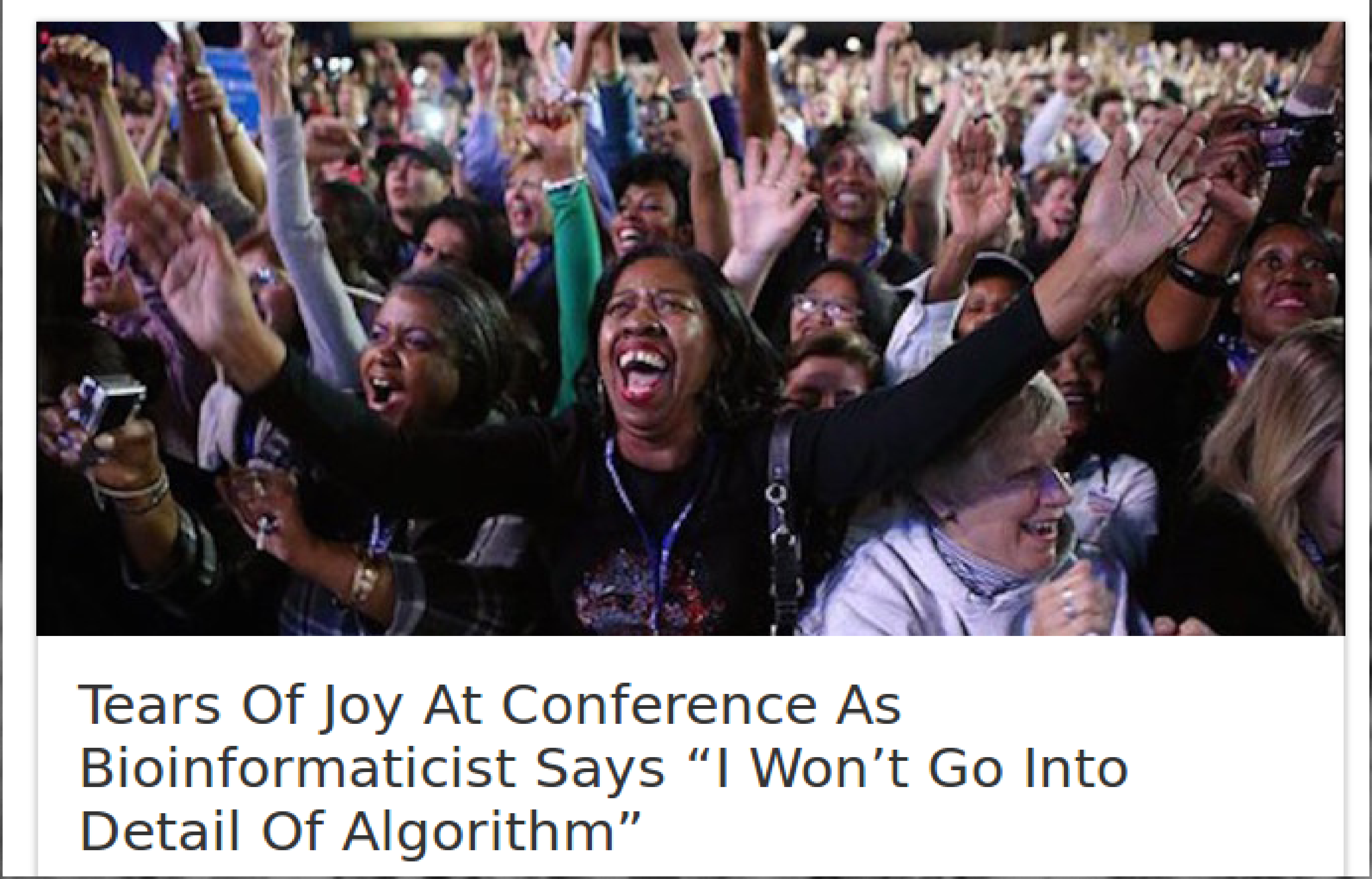
Computational Methods

Bioinformatics activities require the use of custom-developed software.

Behind this, there is algorithm development and statistics, that need to be dealt-with quality concerns.

Users, on the other hand, need to be able to **critically assess** the behaviour and the performance of these tools, in order to trust the results.

A matter of trust...



Tears Of Joy At Conference As Bioinformaticist Says “I Won’t Go Into Detail Of Algorithm”

Bioinformatics Tools, some examples

- Assessing data quality
- Comparing gene expression in large arrays
- Evaluating scores in sequence aligments
- Evaluating fitness of structural models to data
- Evaluating variation in genomic data
- Classifying gene expression patterns
- Classifying population genetics data
- Assessing association and correlation
- Inferring causality in observations and experiments

Bioinformatics Tools, specific needs

- Fitting
- Regression
- Maximization
- Statistical tests
- Correcting for multiple testing
- Machine learning
- Design of experiments
- Testing the performance of classifiers, etc.

Alignments and Searches

Looking for sequence similarity

Alignment

- Global
- Local (BLAST and its variants)

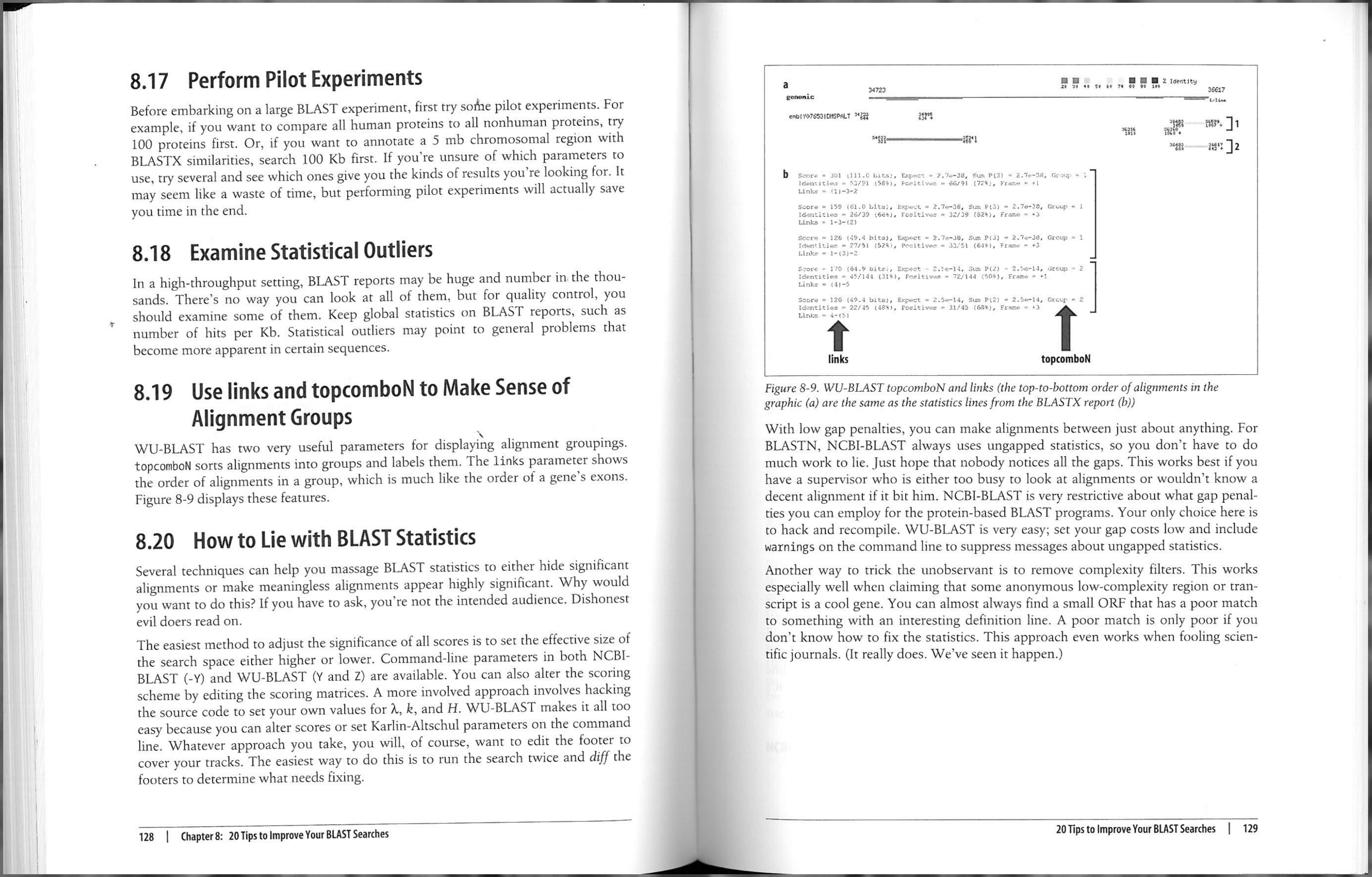
Searching in databases and ranking by similarity

Identification of Motifs and Domains

BLAST statistics

BLAST = Basic Local Alignment Search Tool

In “BLAST” by Ian Korf, Mark Yandell and Joseph Bedell, O'Reilly, 2003



8.17 Perform Pilot Experiments

Before embarking on a large BLAST experiment, first try some pilot experiments. For example, if you want to compare all human proteins to all nonhuman proteins, try 100 proteins first. Or, if you want to annotate a 5 mb chromosomal region with BLASTX similarities, search 100 Kb first. If you're unsure of which parameters to use, try several and see which ones give you the kinds of results you're looking for. It may seem like a waste of time, but performing pilot experiments will actually save you time in the end.

8.18 Examine Statistical Outliers

In a high-throughput setting, BLAST reports may be huge and number in the thousands. There's no way you can look at all of them, but for quality control, you should examine some of them. Keep global statistics on BLAST reports, such as number of hits per Kb. Statistical outliers may point to general problems that become more apparent in certain sequences.

8.19 Use links and topcomboN to Make Sense of Alignment Groups

WU-BLAST has two very useful parameters for displaying alignment groupings. `topcomboN` sorts alignments into groups and labels them. The `links` parameter shows the order of alignments in a group, which is much like the order of a gene's exons. Figure 8-9 displays these features.

8.20 How to Lie with BLAST Statistics

Several techniques can help you massage BLAST statistics to either hide significant alignments or make meaningless alignments appear highly significant. Why would you want to do this? If you have to ask, you're not the intended audience. Dishonest evil doers read on.

The easiest method to adjust the significance of all scores is to set the effective size of the search space either higher or lower. Command-line parameters in both NCBI-BLAST (`-Y`) and WU-BLAST (`Y` and `Z`) are available. You can also alter the scoring scheme by editing the scoring matrices. A more involved approach involves hacking the source code to set your own values for λ, *k*, and *H*. WU-BLAST makes it all too easy because you can alter scores or set Karlin-Altschul parameters on the command line. Whatever approach you take, you will, of course, want to edit the footer to cover your tracks. The easiest way to do this is to run the search twice and *diff* the footers to determine what needs fixing.

```
Score = 301 (111.0 bits), Expect = 2.7e-38, Sum P(3) = 2.7e-38, Group = 1
Identities = 53/91 (58%), Positives = 66/91 (72%), Frame = +1
Links = (1)-3-2

Score = 159 (61.0 bits), Expect = 2.7e-38, Sum P(3) = 2.7e-38, Group = 1
Identities = 26/39 (66%), Positives = 32/39 (82%), Frame = +3
Links = 1-3-(2)

Score = 126 (49.4 bits), Expect = 2.7e-38, Sum P(3) = 2.7e-38, Group = 1
Identities = 27/51 (52%), Positives = 33/51 (64%), Frame = +3
Links = 1-(3)-2

Score = 170 (64.9 bits), Expect = 2.5e-14, Sum P(2) = 2.5e-14, Group = 2
Identities = 45/144 (31%), Positives = 72/144 (50%), Frame = +1
Links = (4)-5

Score = 126 (49.4 bits), Expect = 2.5e-14, Sum P(2) = 2.5e-14, Group = 2
Identities = 22/45 (48%), Positives = 31/45 (68%), Frame = +3
Links = 4-(5)
```

Figure 8-9. WU-BLAST topcomboN and links (the top-to-bottom order of alignments in the graphic (a) are the same as the statistics lines from the BLASTX report (b))

With low gap penalties, you can make alignments between just about anything. For BLASTN, NCBI-BLAST always uses ungapped statistics, so you don't have to do much work to lie. Just hope that nobody notices all the gaps. This works best if you have a supervisor who is either too busy to look at alignments or wouldn't know a decent alignment if it bit him. NCBI-BLAST is very restrictive about what gap penalties you can employ for the protein-based BLAST programs. Your only choice here is to hack and recompile. WU-BLAST is very easy; set your gap costs low and include `warnings` on the command line to suppress messages about ungapped statistics.

Another way to trick the unobservant is to remove complexity filters. This works especially well when claiming that some anonymous low-complexity region or transcript is a cool gene. You can almost always find a small ORF that has a poor match to something with an interesting definition line. A poor match is only poor if you don't know how to fix the statistics. This approach even works when fooling scientific journals. (It really does. We've seen it happen.)

Prediction and Inference

Predicting structure from sequence

- Sequence motifs
- Domains as sets of motifs
- Fingerprints

Inferring function from sequence and structure

- By similarity (holomogy modelling)

Acquiring confidence in results

Bioinformatics can produce results that are not **consistent**.

Often the question is pushing us back into the statistical methods to try to isolate the causes of inconsistency.

We also need to guarantee that the results are fully **reproducible**.

The need for **knowledge** in statistics is just everywhere.

Education and Training

Bioinformatics requires knowledge in Statistics at a variety of levels, naturally different for **researchers**, **software developers** and **users**.

Formal learning (higher education, degrees)

Non-formal learning

- Training
- Continuing Professional Education (CPE)

Education

Revision of undergraduate curricula in Statistics

Possible enhancement via **online learning** and **flipped class** techniques

Possible enhancement via **Peer Instruction**

“A New Online Computational Biology Curriculum” by David Searls, PLOS Comp. Bio. **10**, 6 June 2014.

From “Computational Biology Online Course Catalog”:

https://www.coursera.org/course/statistics

http://ocw.jhsph.edu Methods in Biostatistics I

Biostatistics Training in GTPB

Meeting the needs of:
End users
Developers
Researchers
other trainers (instructors)

GTPB

The Gulbenkian Training Programme in Bioinformatics
(Since 1999)

Pedro Fernandes, Organiser

Introductory Biostatistics for Biologists

The importance of Statistics.
Quantitative observations. Accuracy and Precision. Observations with error. Chance. Probabilities. Causation.

Descriptive Statistics. Basic concepts.
Describing and summarizing data. Summary statistics and plots for univariate and bivariate data.

Review of probability theory
Probability, random variables and their properties.
Independence and conditional probability.
Distributions: discrete random variables and continuous random variables.

Statistical inference
Sampling distributions. Confidence Intervals.
Hypothesis testing (parametric tests).

Statistical inference
Hypothesis testing (non-parametric tests). Contingency tables.
Design of experiments
ANOVA: one-way, two-way, repeated measures.
Factorial design. Latin Squares.

Advanced Biostatistics for Biologists

Significance and p-value
Multiple testing issues
Corrections for multiple testing.

Simulation modelling methodologies
Monte Carlo and Bootstrap methods
Parametric approach, Non-Parametric approach

Bayesian inference
Bayes' theorem. Principles of Bayesian methodology. Gibbs Sampling.
Statistical inference. Expectation-Maximization (EM) algorithm.

Multivariate data analysis
Organising multivariate data.
Principal component analysis.

Machine Learning in Bioinformatics.
Statistical Methods for NGS Data Analysis
Introduction. Using the EDASeq package.
Using the edgeR, DESeq packages.

Training in GTPB

The two training courses are designed to meet strict learning objectives

The content tends to compensate deficiencies in basic knowledge

Results are quite rewarding but the effort is higher than normal

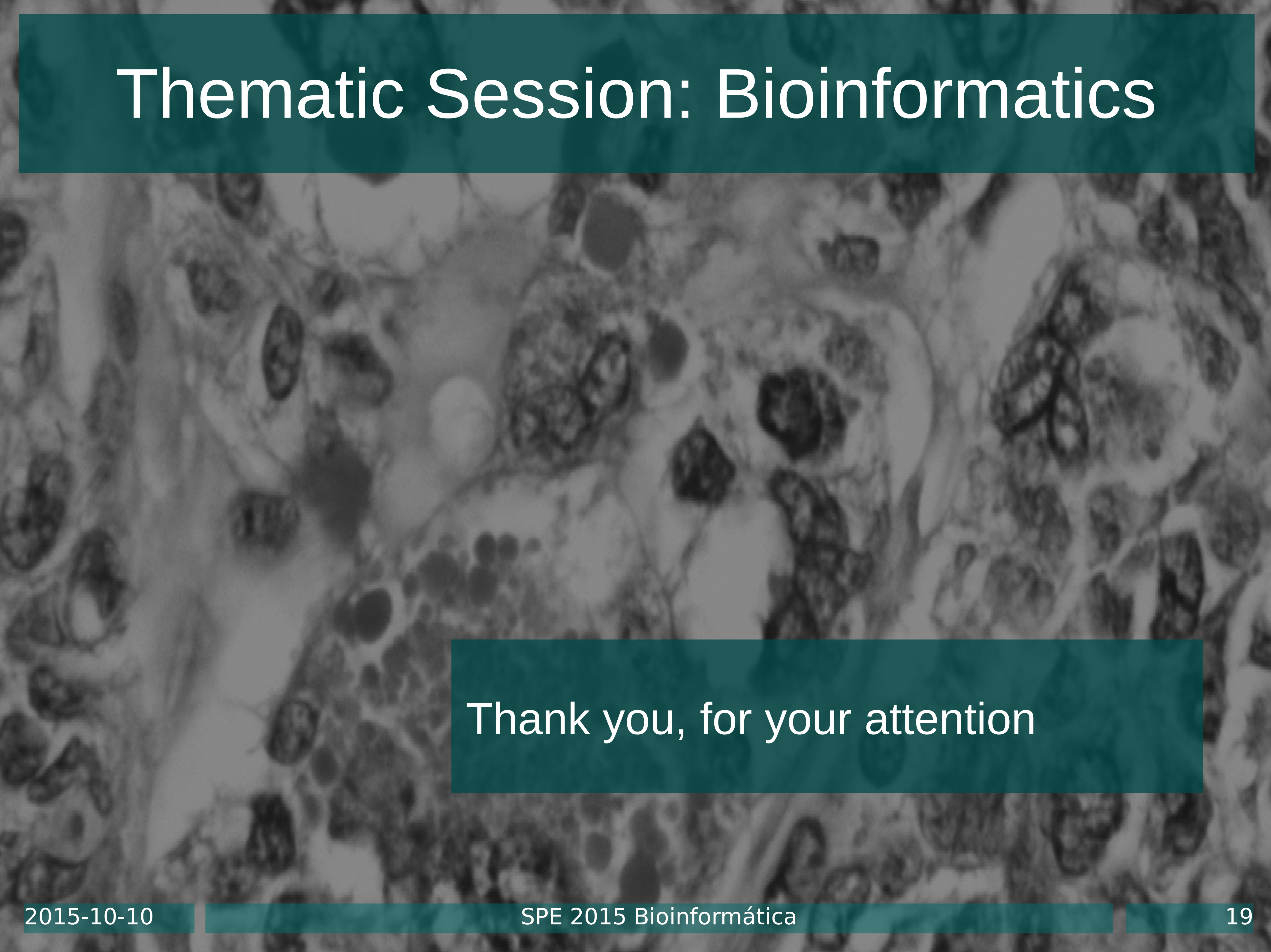
Attempts will be made to offset basic subjects by using online content

Acknowledgements

Acknowledgements are due to several hundred collaborators, who have accompanied me in providing training to thousands of Bioinformatics users and developers, among which I have to cite the following statisticians for their very generous contributions

Maria Antónia Turkman
Lisete de Sousa
Carina Silva
Ana Luísa Papoila
Maria Fernanda Diamantino

Thematic Session: Bioinformatics



Thank you, for your attention