What mathematical optimization can, and cannot, do for biologists

Steven Kelk
Department of Knowledge Engineering (DKE)
Maastricht University, NL

Introduction

- There is no shortage of literature about the role of mathematics and computer science in computational biology.
- The bottom line is well-known: computers, and the software and algorithms that run on them, have become a fundamental part of modern biology.
- However, for people not working in mathematics or computer science, it remains rather “mystical” how software and algorithms are developed, and what they can - and cannot - do.

Introduction

- In this talk I want to try and de-mystify these issues a little, because I think this is important for the effective use of computers in biology.
- I'll do this by drawing on my own experiences in computational biology.
- Particularly inspirational are those moments when you realise that, outside your own corner of expertise, very few concepts are "obvious" or "universal"…

A disclaimer…

- I'm a discrete mathematician, mainly interested in combinatorial optimization, so there will inevitably be a bias in that direction during this talk.

- In particular, I feel a bit guilty that I have not said more about statistical methods. These methods are extremely important in computational biology, and phylogenetics is no exception.

- On the other hand, the principles of mathematical optimization often apply here too. For example, *"How do I compute the maximum likelihood tree?"* or`*"How long should I run my Monte Carlo Markov Chain to guarantee that it is close to its uniform distribution?"*

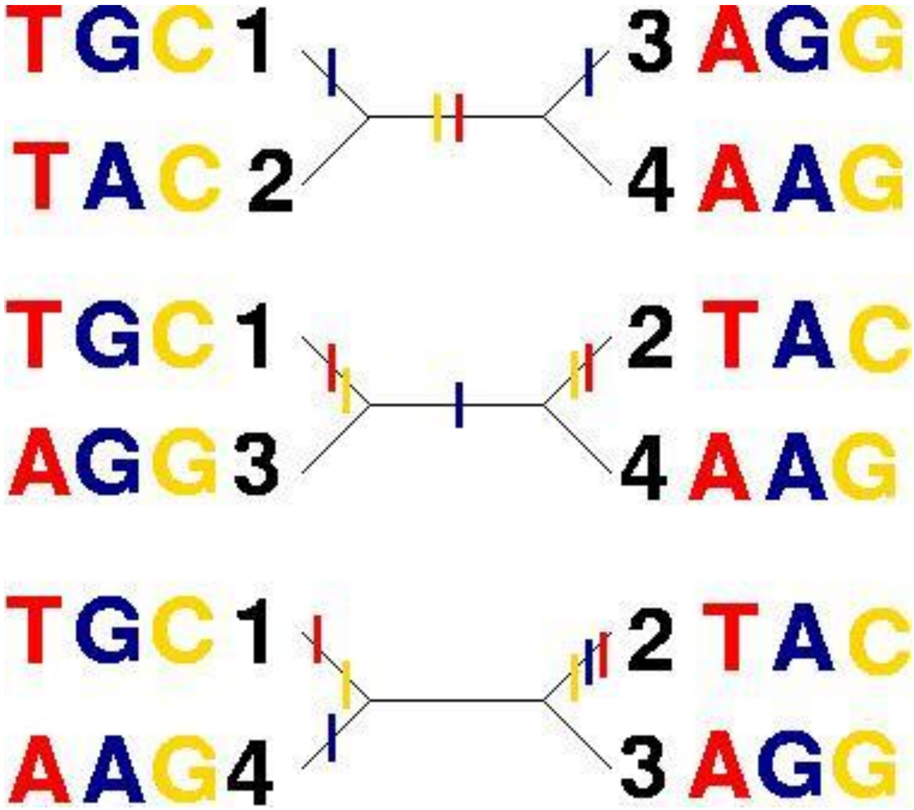
Two motivating examples

Maximum Parsimony (MP)

Sequence 1	T	G	C
Sequence 2	T	A	C
Sequence 3	A	G	G
Sequence 4	A	A	G

Input

Space of feasible solutions

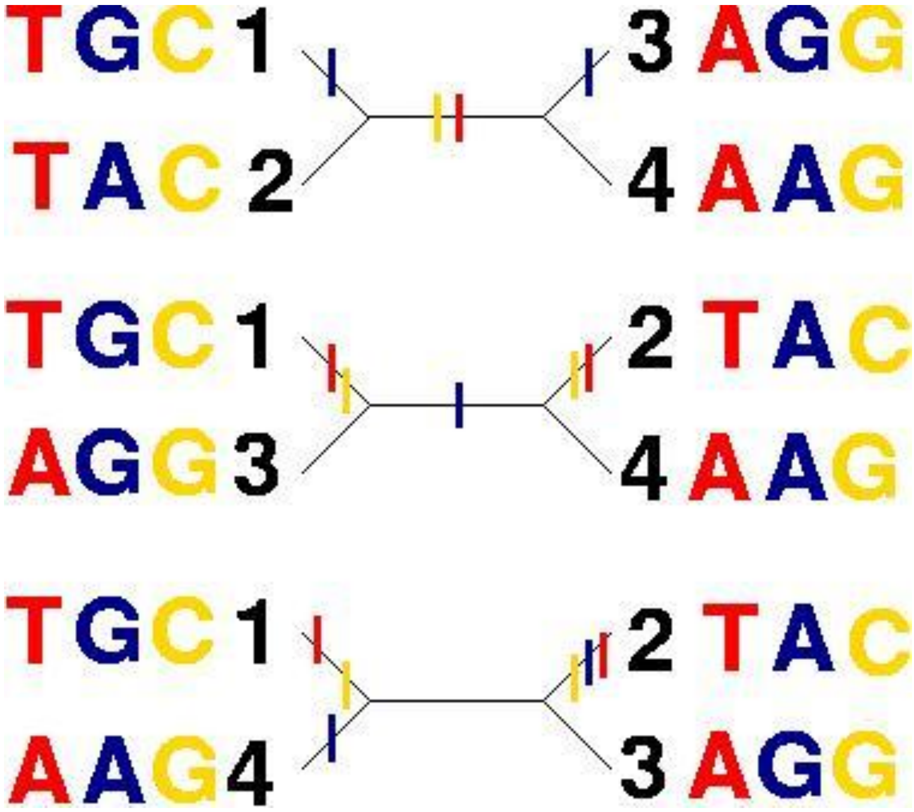


From: http://artedi.ebc.uu.se/course/X3-2004/Phylogeny/Exercises/mp.html

Maximum Parsimony (MP)

Sequence 1	T	G	C
Sequence 2	T	A	C
Sequence 3	A	G	G
Sequence 4	A	A	G

Input



Space of feasible solutions

What is the "best" solution?

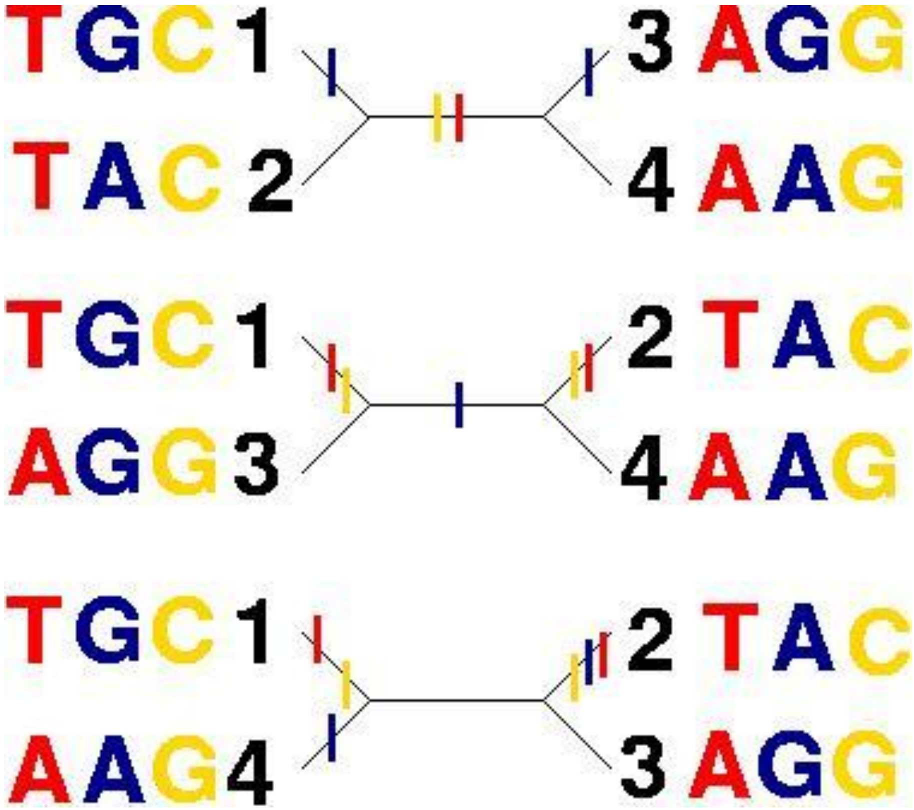
From: http://artedi.ebc.uu.se/course/X3-2004/Phylogeny/Exercises/mp.html

Maximum Parsimony (MP)

Sequence 1	T	G	C
Sequence 2	T	A	C
Sequence 3	A	G	G
Sequence 4	A	A	G

Input

Space of feasible solutions



Before we can determine the “best” tree, we need to formalize what “best” means…

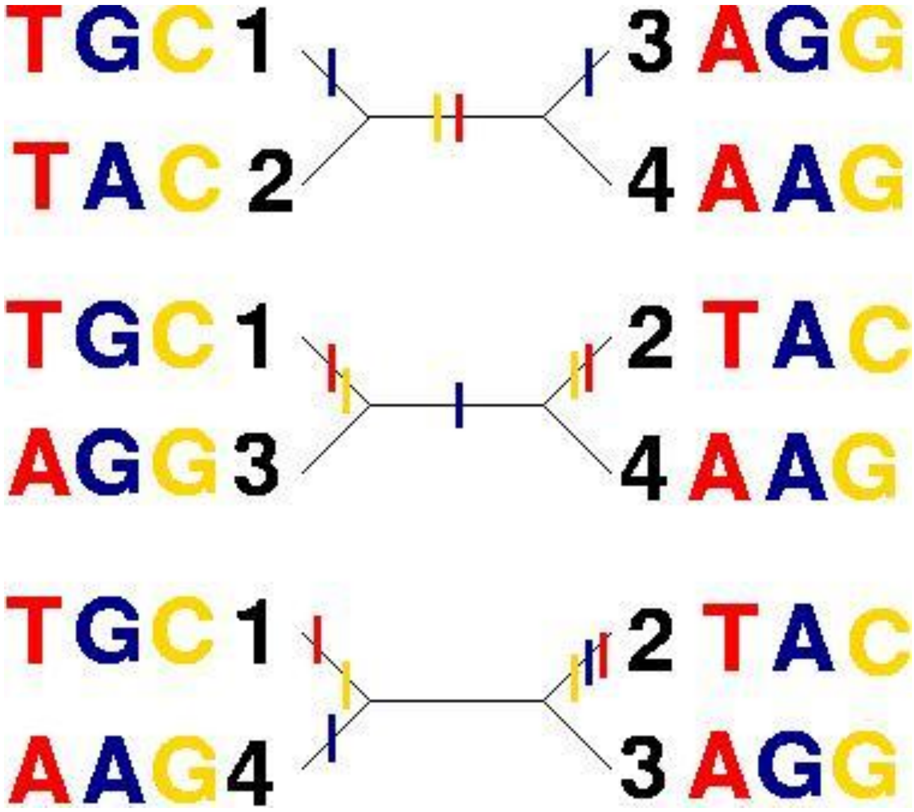
From: http://artedi.ebc.uu.se/course/X3-2004/Phylogeny/Exercises/mp.html

Maximum Parsimony (MP)

Sequence 1	T	G	C
Sequence 2	T	A	C
Sequence 3	A	G	G
Sequence 4	A	A	G

Input

Space of feasible solutions



In MP, the “quality” of a tree is the number of mutations along its edges.

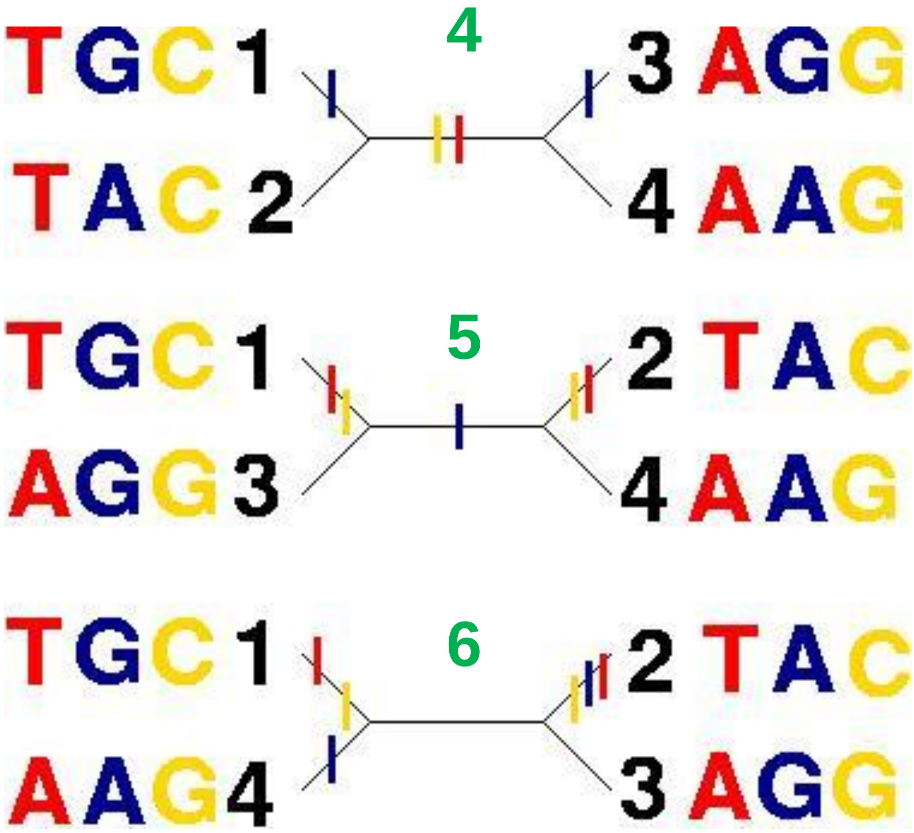
The “best” tree is the one that minimizes this number.

From: http://artedi.ebc.uu.se/course/X3-2004/Phylogeny/Exercises/mp.html

Maximum Parsimony (MP)

Sequence 1	T	G	C
Sequence 2	T	A	C
Sequence 3	A	G	G
Sequence 4	A	A	G

Input

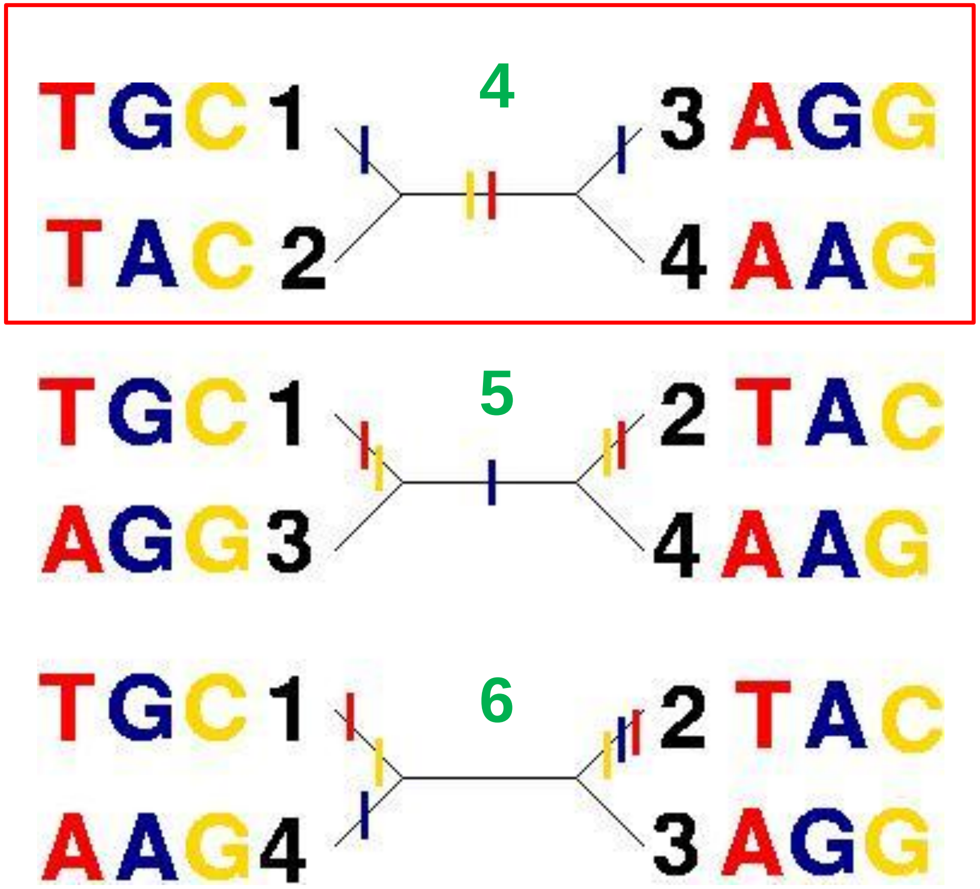


From: http://artedi.ebc.uu.se/course/X3-2004/Phylogeny/Exercises/mp.html

Sequence 1	T	G	C
Sequence 2	T	A	C
Sequence 3	A	G	G
Sequence 4	A	A	G

Input

Maximum Parsimony (MP)



So the **optimal** tree is the first one (i.e. fewest mutations), with a parsimony score of 4.

Here the word “optimal” says **nothing** about the biological relevance of the tree. It simply means: the tree that minimizes the parsimony score.

From: http://artedi.ebc.uu.se/course/X3-2004/Phylogeny/Exercises/mp.html

MP as an optimization problem

- **Input:** a set of *n* sequences

- **Space of feasible solutions:** all unrooted phylogenetic trees on *n* leaves, where the leaves are labelled by the input sequences

- **Objective function:** the quality of a tree is defined to be its parsimony score i.e. the number of mutations along its edges

- **Goal:** find a tree that minimizes the objective function. This is the **optimal** tree.

MP as an optimization problem

- Note that this does not say anything about how the optimal tree should be constructed, or even if it is computationally realistic.

- In an ideal world, there would be a perfect separation between the mathematical model (which is supposed to be an approximation of biological reality), and the question of how to efficiently find the optimal solution within the mathematical model.

- In practice mathematical models are heavily influenced by the limits of computation (“tractability”). More about this later.

Maximum Likelihood (ML) as an optimization problem

- **Input:** a set of *n* sequences and a probability distribution on nucleotide mutations

- **Space of feasible solutions:** all unrooted phylogenetic trees on *n* leaves, where the leaves are labelled by the input sequences

- **Objective function:** the quality of a tree is defined to be the likelihood of observing that tree given the input probability distribution

- **Goal:** find a tree that maximizes the objective function. This is the **optimal** tree.

Abstraction

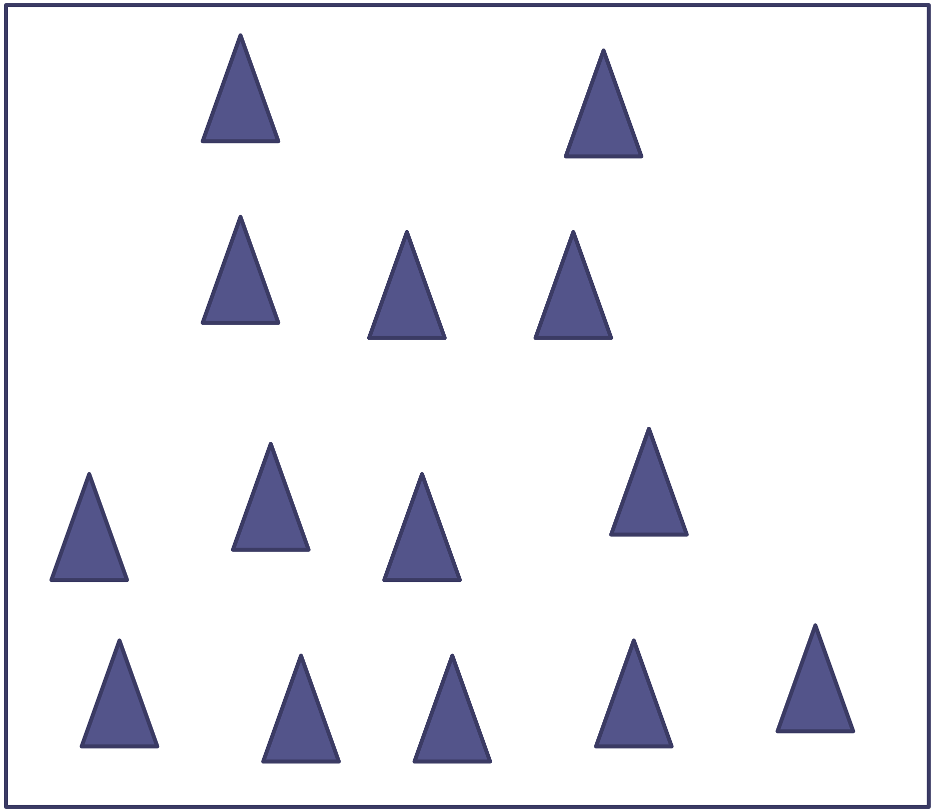
Input (e.g.character data,
distance data,
trees, networks)

Objective function (e.g.
parsimony, likelihood,
minimizing reticulation)

Space of feasible solutions (e.g.
space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

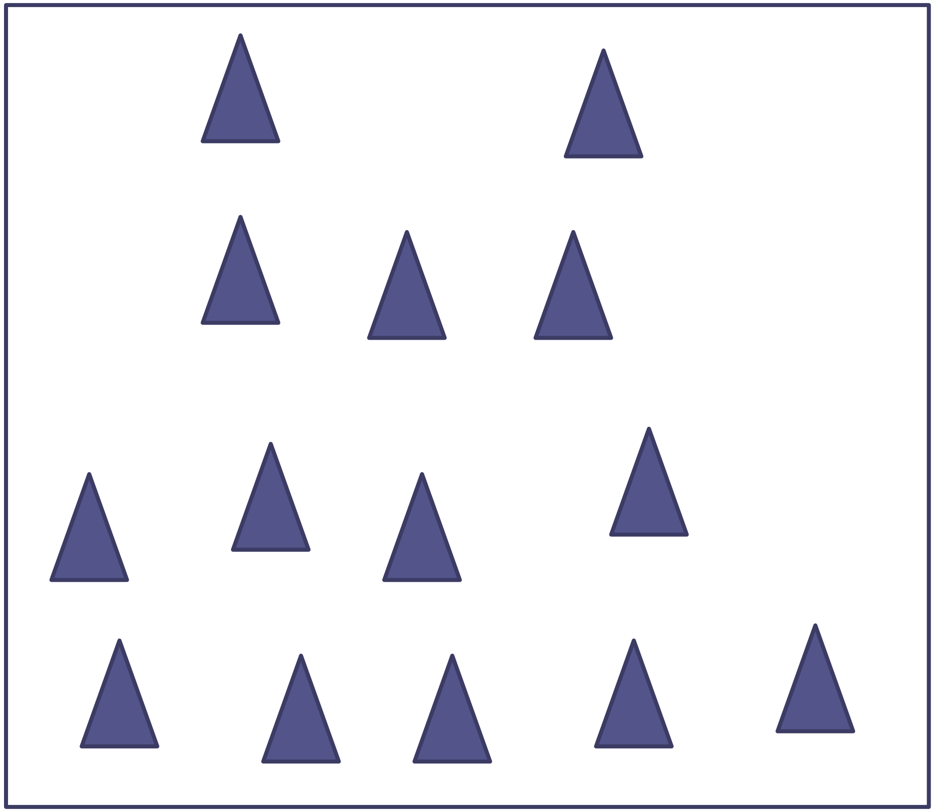


Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data,
distance data,
trees, networks)

Objective function (e.g.
parsimony, likelihood,
minimizing reticulation)

The objective function imposes a hierarchy (i.e. an ordering) on the space of feasible solutions



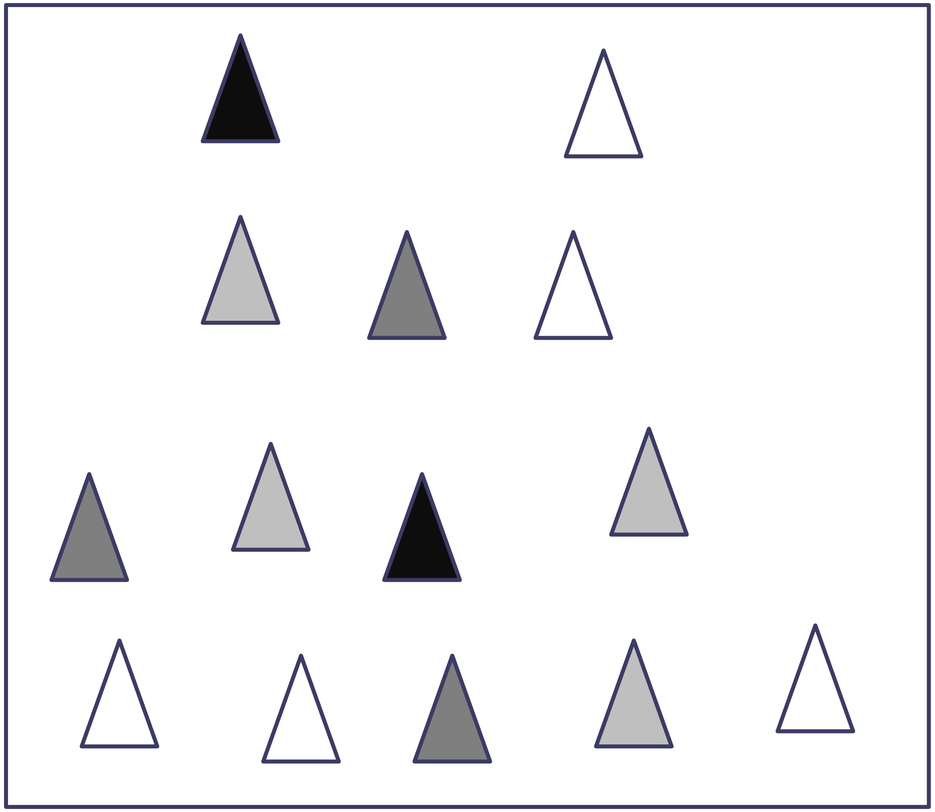
Space of feasible solutions (e.g.
space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

The objective function imposes a hierarchy (i.e. an ordering) on the space of feasible solutions.

Here: darker = better

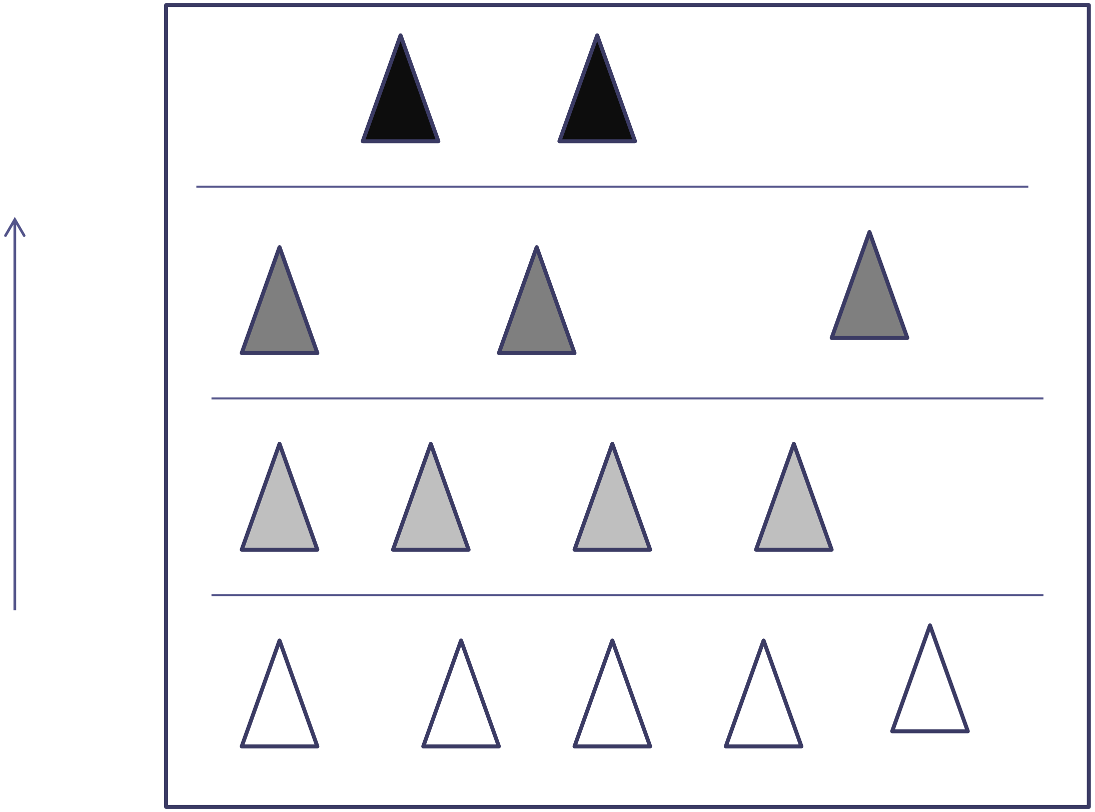


Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data,
distance data,
trees, networks)

Objective function (e.g.
parsimony, likelihood,
minimizing reticulation)

Solutions
grouped
according to
increasing
quality



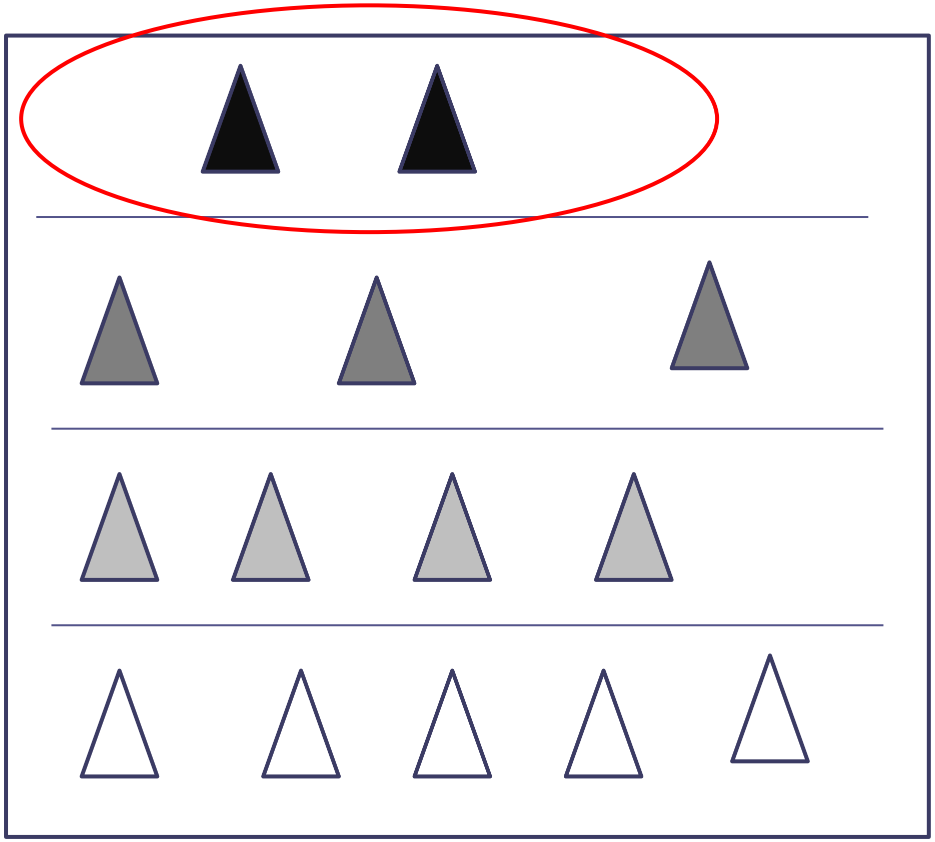
Space of feasible solutions (e.g.
space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

There are two optimal solutions. So in this case there is **no unique** optimal solution.

It can be very dangerous to make biological inferences based on seeing only a **single** optimal solution!



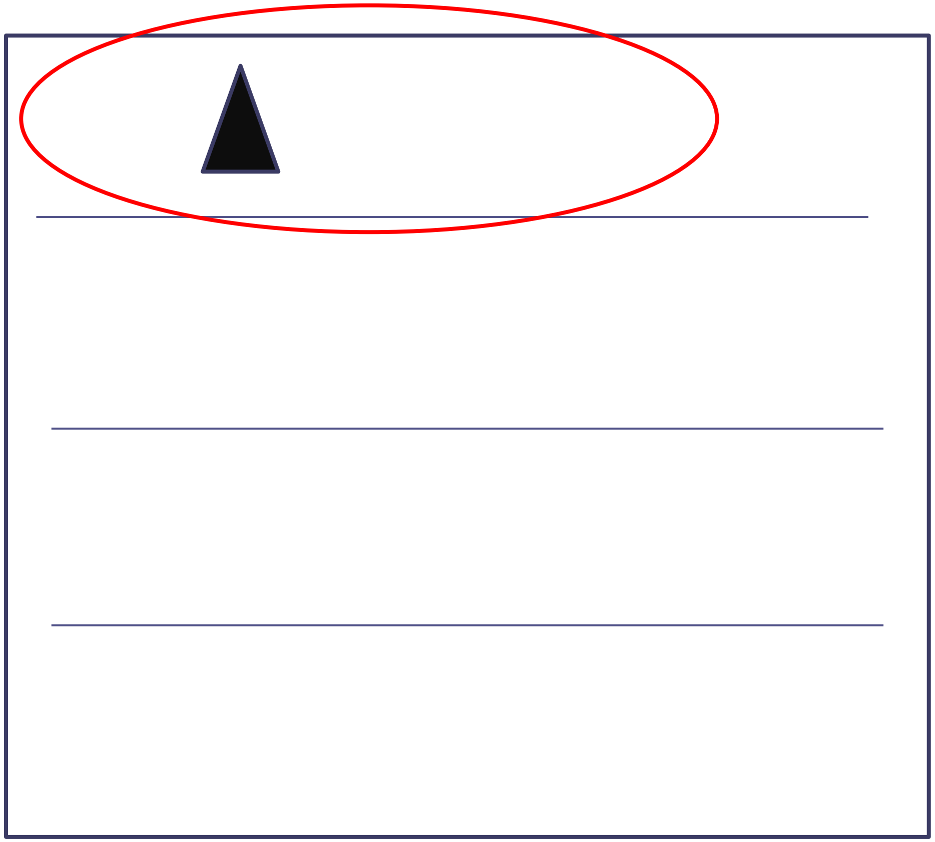
Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

There are two optimal solutions. So in this case there is **no unique** optimal solution.

It can be very dangerous to make biological inferences based on seeing only a **single** optimal solution!



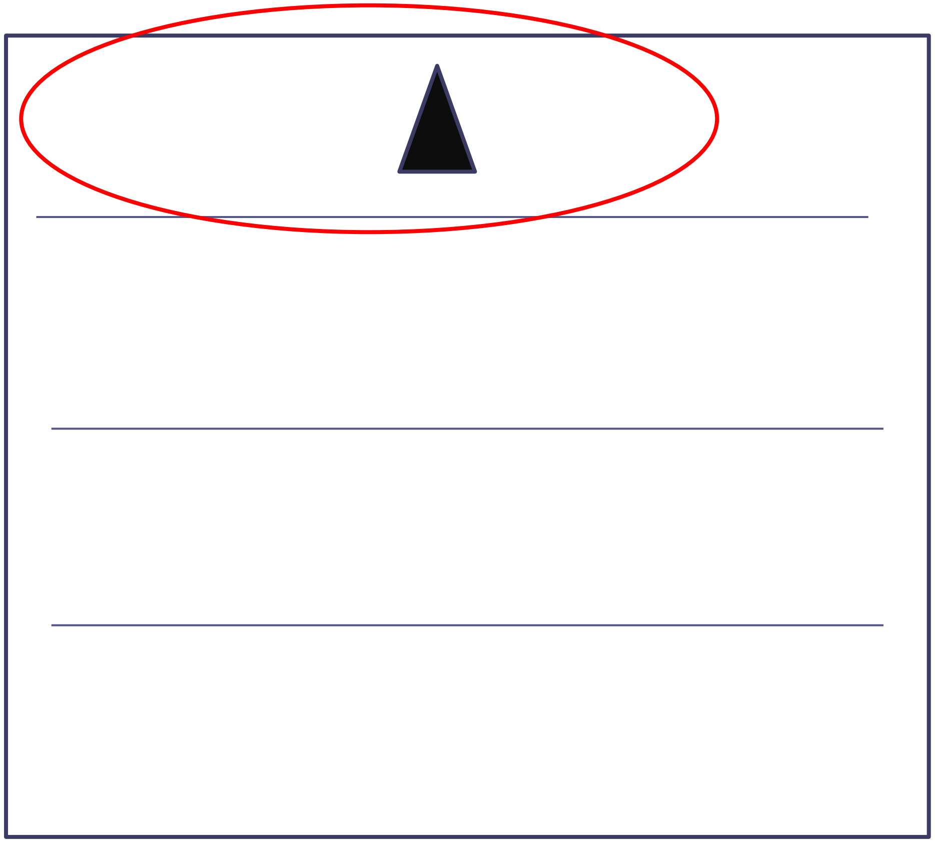
Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

There are two optimal solutions. So in this case there is **no unique** optimal solution.

It can be very dangerous to make biological inferences based on seeing only a **single** optimal solution!



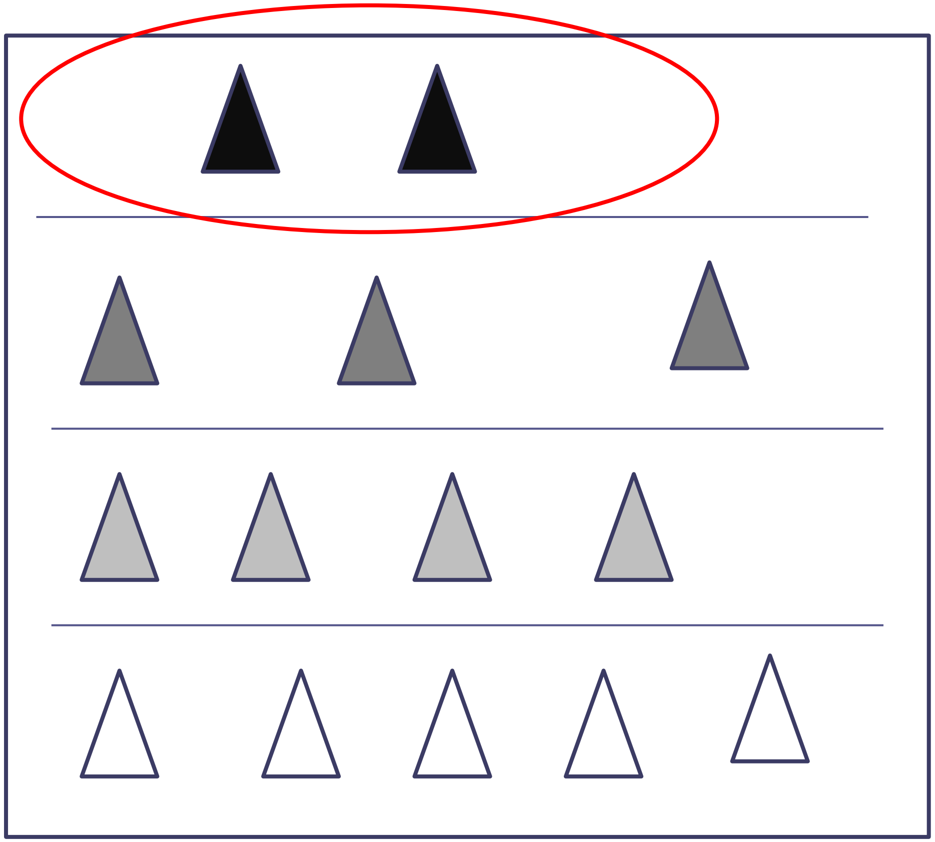
Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

There are two optimal solutions. So in this case there is **no unique** optimal solution.

Ideally we want software to accurately describe the common characteristics of **all** optimal solutions.



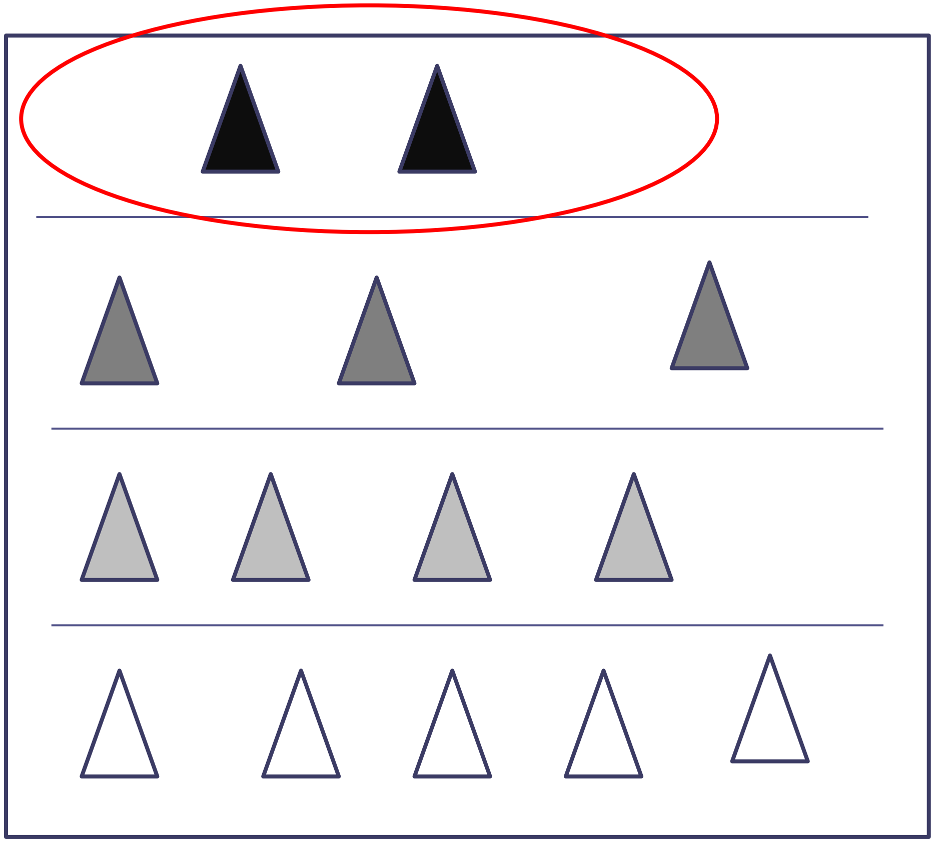
Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

There are two optimal solutions. So in this case there is **no unique** optimal solution.

Unfortunately, in many cases finding even **one** optimal (or near-optimal) solution is already a major computational challenge.



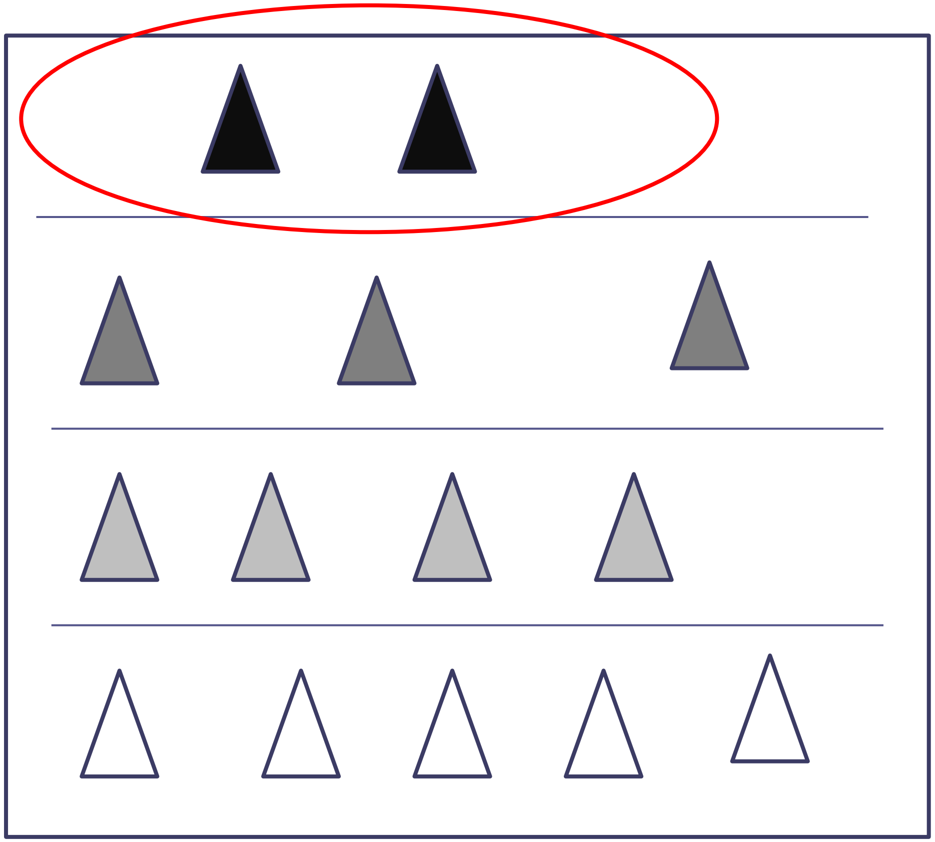
Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

There are two optimal solutions. So in this case there is **no unique** optimal solution.

Also, there is often also a **vast** number of optimal solutions, and summarizing them is also computationally problematic (due to **symmetries**).



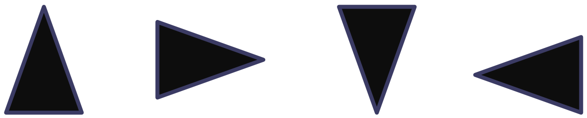
Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

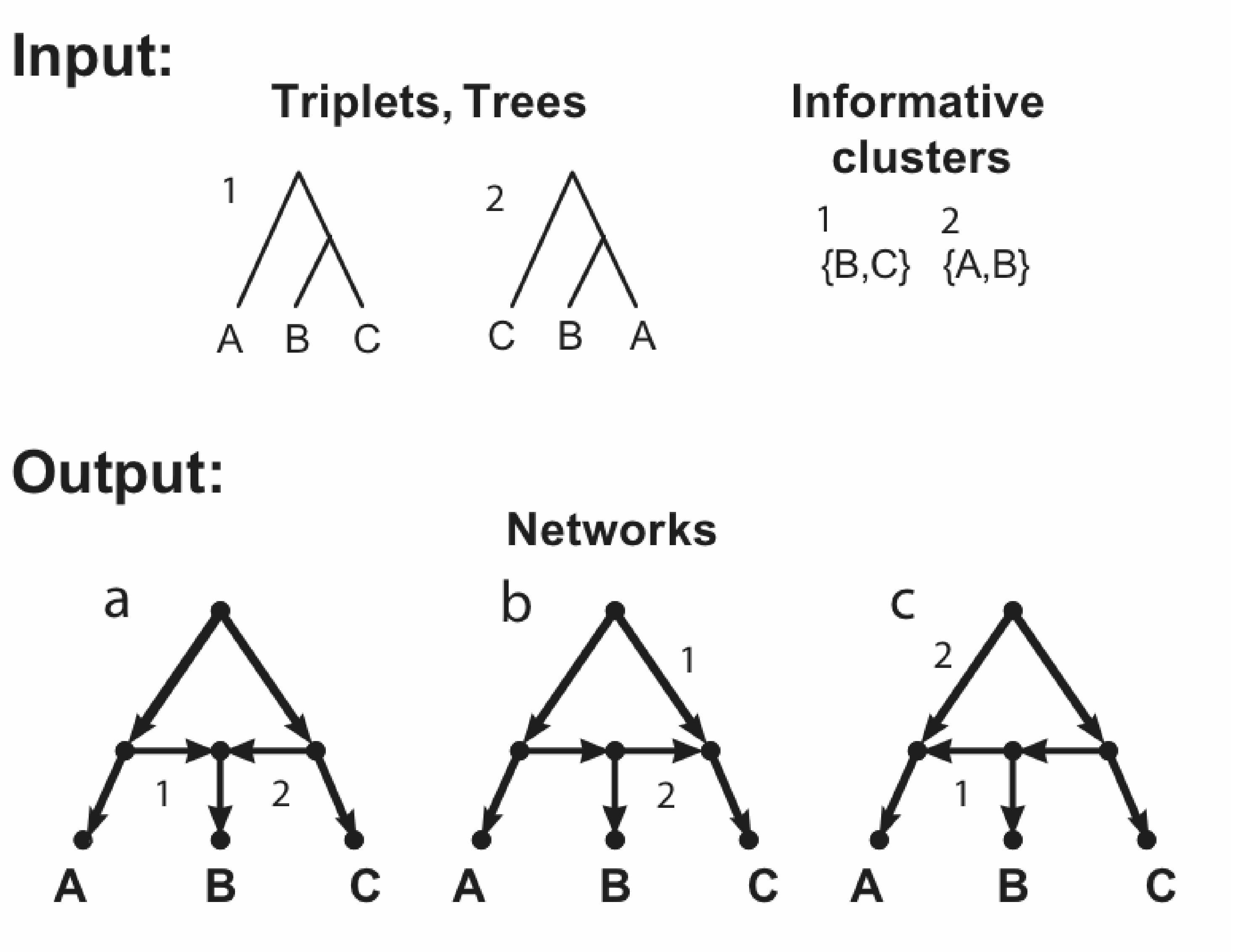
There are two optimal solutions. So in this case there is **no unique** optimal solution.

Also, there is often also a **vast** number of optimal solutions, and summarizing them is also computationally problematic (due to **symmetries**).



Four optimal solutions that are mathematically distinct, but are in fact trivial symmetries of each other – can distort enumeration and sampling algorithms

Space of feasible solutions (e.g. space of trees, networks)



Input:

Triplets, Trees

Informative clusters

1 {B,C} 2 {A,B}

Output:

Networks

http://phylonetworks.blogspot.nl/2012_02_01_archive.html

Some modelling issues

Modelling issues (1) – every cloud has a silver lining

- The exact form of the input, the objective function, and the space of feasible solutions is a **modelling choice** that should be made very carefully.

- Due to computational limitations, mathematicians often encourage biologists to make the model as simple, and well-defined, as possible.

- This attempt to limit model complexity is pragmatic, and restrictive. But it arguably has a positive side too.

Modelling issues (1) – every cloud has a silver lining

- It forces us to ask ourselves what “biologically plausible” really means (i.e. to identify implicit assumptions).

- The modelling phase is also a good moment to clarify whether software is being used for hypothesis *generation*, or for hypothesis *testing*: a subtle but important distinction.

Modelling issues (2) – if nothing else, lower bounds

- Some mathematical optimization models (e.g. Maximum Likelihood) are considered to be quite accurate at reconstructing phylogenies.
- This is based on many years of practice and validation.
- Without that level of validation it is dangerous to use a mathematical optimization model to draw direct biological conclusions.
- But even in the absence of validation, mathematical optimization models can give useful “lower bounds” on the complexity of the hypothesis required to explain the input data (i.e. the observed phenomena). MP is a classic example of this.

Modelling issues (3) – Mixed-up messages

- An (at the time, for me unexpected) danger of mathematical modelling is nicely illustrated by the following.

- In the rooted phylogenetic networks community, there has been quite a bit of interest from mathematicians in “level-k” networks.

- The level of a network is just a measurement of how locally reticulate it is. The higher the level, the more reticulate the network is. It is just a measurement, **not a value judgement on the plausibility of the network.**

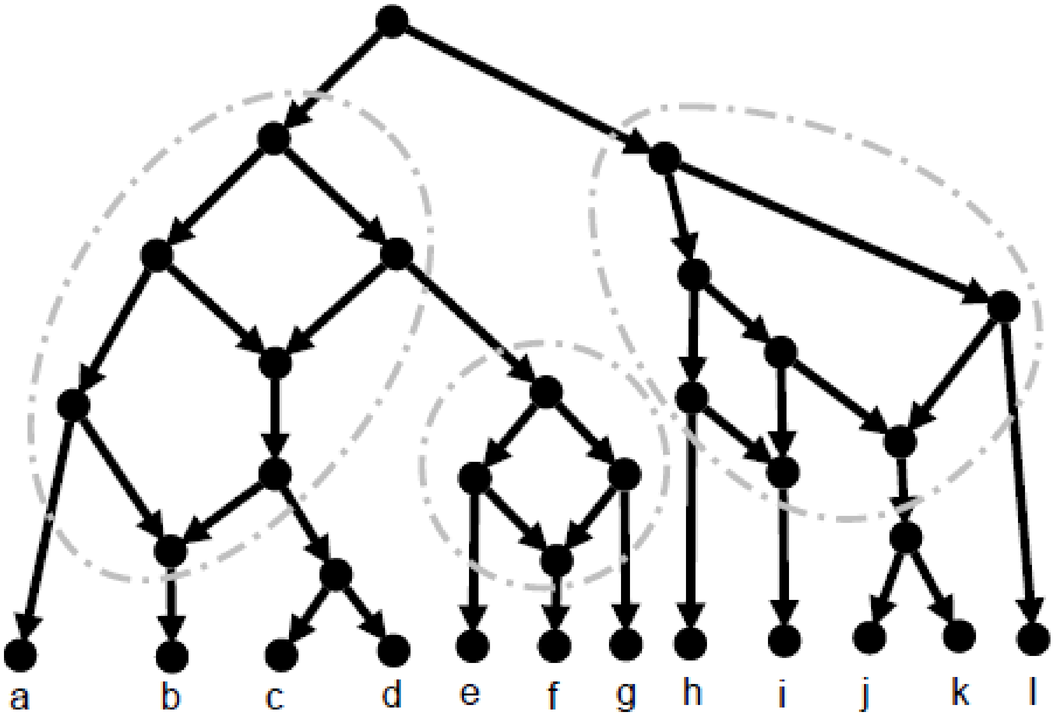


Fig. 1 Example of a phylogenetic network with five reticulations. The encircled subgraphs form its biconnected components, also known as its "tangles". This binary network has level equal to 2 since each biconnected component contains at most two reticulations.

Modelling issues (3) – Mixed-up messages

- One of the reasons why mathematicians liked this way of ordering the space of phylogenetic networks, is that it is computationally easier to optimize over the space of lower-level networks, than higher-level networks.

- This became “hip” and people started asking, *but is it biologically plausible to assume that real phylogenetic networks are low level?*

- But this was never the point: level was intended as a “roadmap” to chart and understand the space of phylogenetic networks. Not as a hypothesis of biological relevance. But this is how it was interpreted!

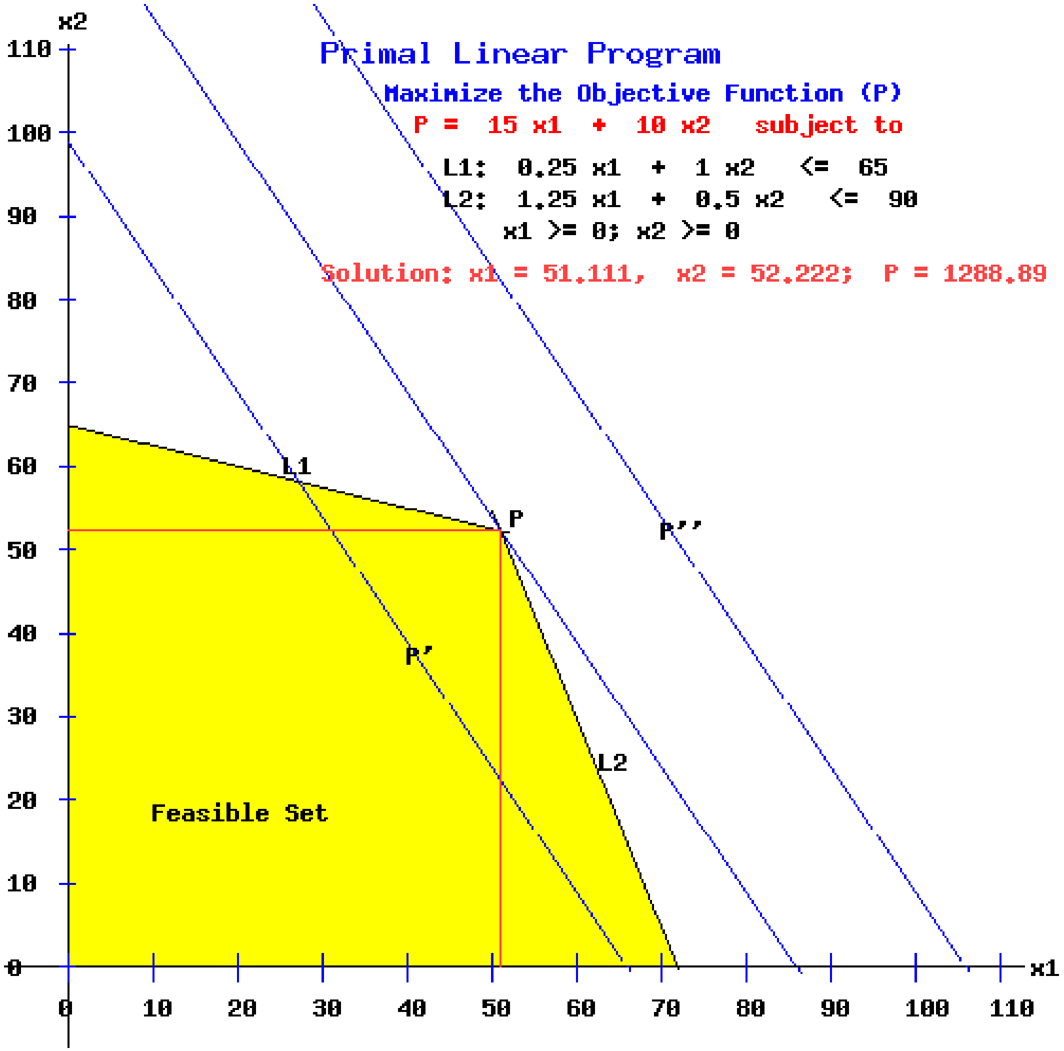
Attaining optimality

“Tractability”

- Suppose we have chosen a mathematical optimization model.
- How do we find an optimal solution? Is it even possible to compute an optimal solution without using an impossible amount of time or computer memory?
- That depends on many factors. In applied mathematics, an enormous amount of research time is spent trying to understand which mathematical optimization problems are **“tractable”** ☺ and which are **“intractable”** ☹.

“Tractability”

- The good news is that some mathematical optimization models are tractable. That is, you can construct optimal solutions quite quickly and without using a huge amount of computer memory.
- There was big progress on this front in the 1950s-1980s.
- Arguably the biggest breakthrough was the realisation that any mathematical optimization model that could be expressed as a **linear program**, could in practice be solved efficiently.



Primal Linear Program

Maximize the Objective Function (P)

P = 15 x1 + 10 x2 subject to

L1: 0.25 x1 + 1 x2 <= 65

L2: 1.25 x1 + 0.5 x2 <= 90

x1 >= 0; x2 >= 0

Solution: x1 = 51.111, x2 = 52.222; P = 1288.89

http://www.egwald.ca/operationsresearch/lpgraphical.php

“Intractability”

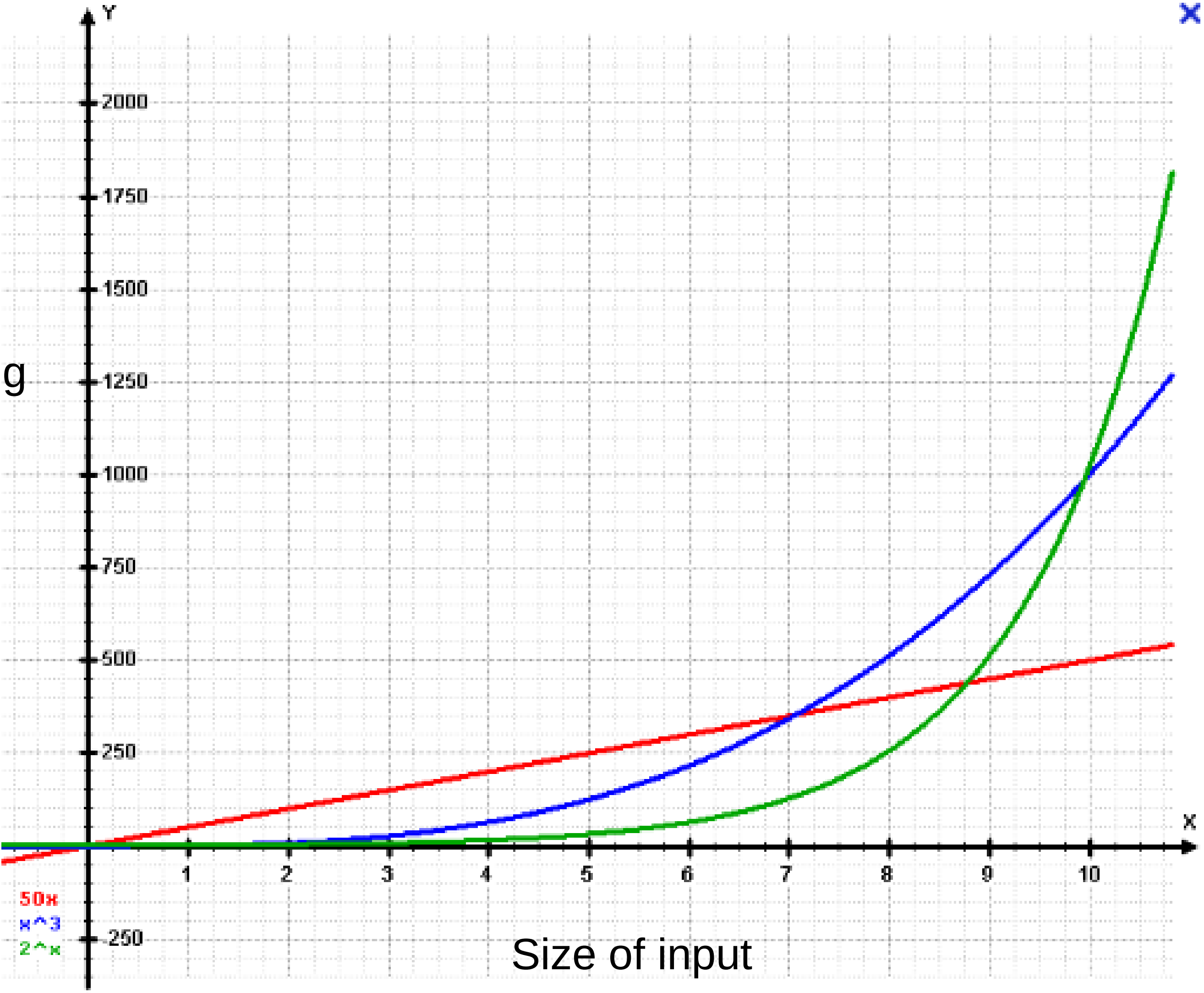
- The bottom line is that if a biological model can be squeezed into a tractable mathematical form that we understand well (e.g. linear programming), then this **is great news**.

- Unfortunately, a great many mathematical models that arise in practice are – at least in theory – **intractable.** This means that the amount of computational time required to construct optimal solutions, grows **explosively** as the size of the input data increases.

- The most well-known way of showing that a problem is intractable, is to give a mathematical proof that it is **“NP-hard”.**

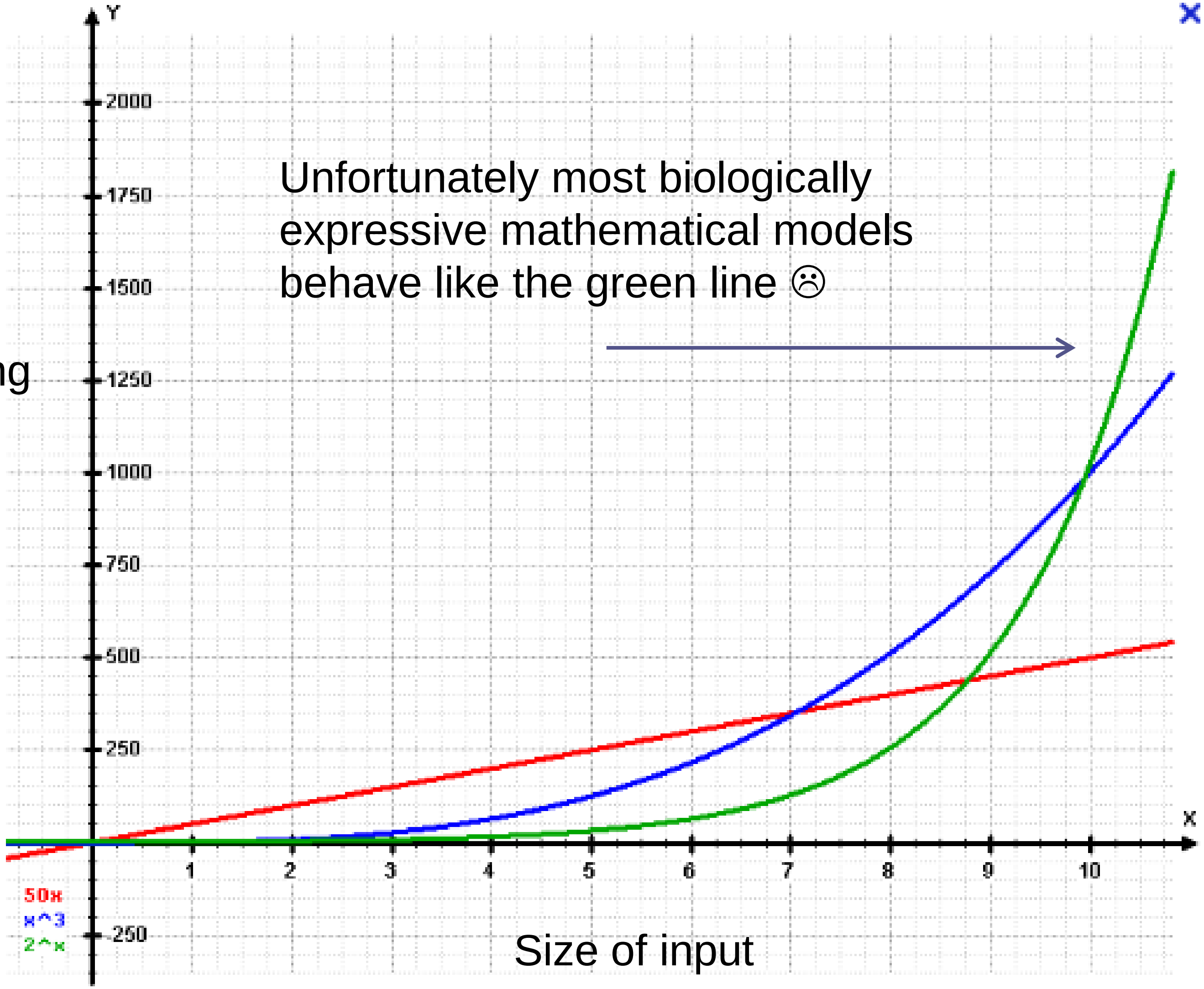
- Models such as MP and ML actually belong to this group.

Running time



Size of input

http://maretbccalculus2007-2008.pbworks.com



Unfortunately most biologically expressive mathematical models behave like the green line ☹

Running time

Size of input

http://maretbccalculus2007-2008.pbworks.com

“Intractability”

- So what should we do if a mathematical optimization model is intractable? Is it the end of the road?
- No. It’s actually only the *beginning* of the story.
- We are helped by the fact that intractability is a **worst-case, “pessimistic”** concept. That is, in many cases it will still be possible to compute optimal solutions.
- There is a huge amount of interest (both research and applied) in developing general software tools that, within limits, can compute optimal solutions to intractable problems. E.g. **Integer Linear Programming (ILP).**

ILOG CPLEX

Documentation

Integer Programming for NP-hard Phylogenetic (and Population- Genetic) Problems

D. Gusfield

Isaac Newton Institute

September 4, 2007

“Intractability” - suboptimality

- Even with the availability of very powerful general tools for tackling intractable problems (such as ILP), there is a limit to what can be done.

- Sometimes you simply have to give up, and accept sub-optimal solutions.

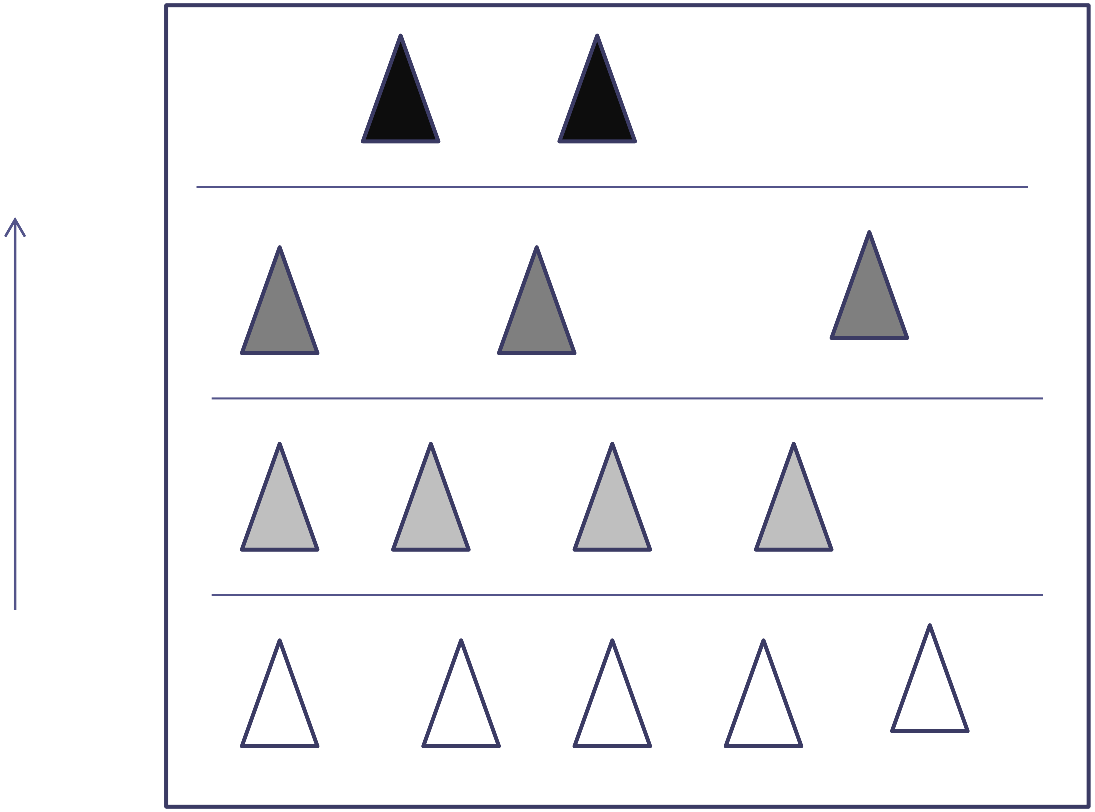
- The good news is that if we no longer insist on finding optimal solutions, it is always possible to find *some* solution.

- The obvious question is then: is it a good solution? This is the fundamental difference between **heuristics** and **approximation algorithms**.

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

Solutions grouped according to increasing quality

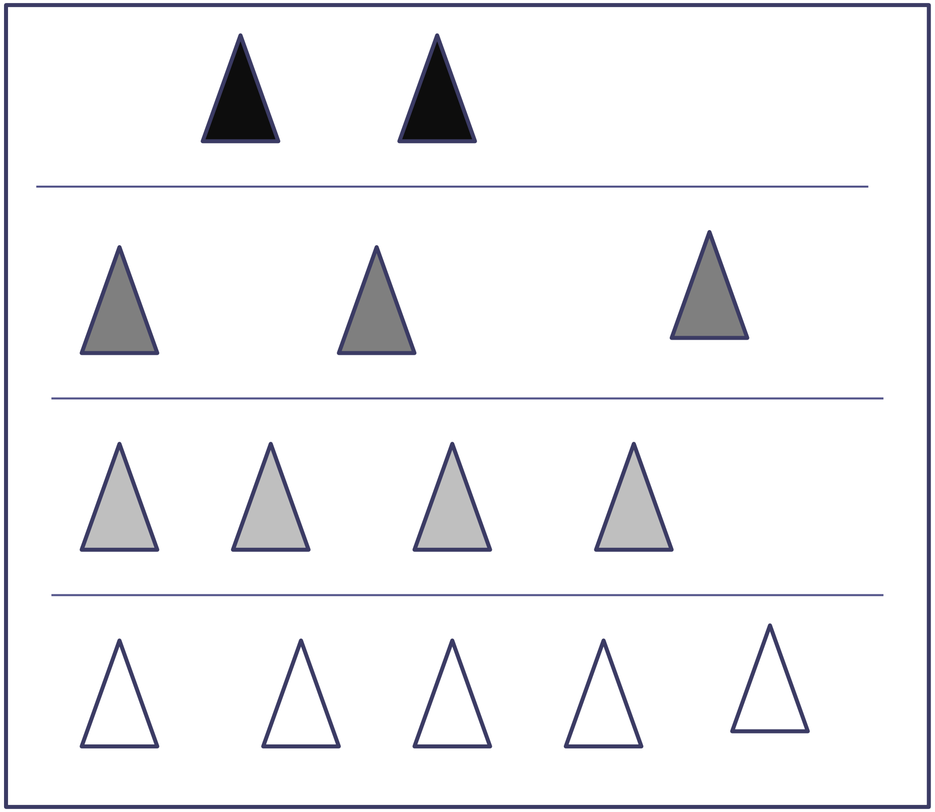


Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

A **heuristic** will give you a solution, but conveys no information about how far the solution is from optimality

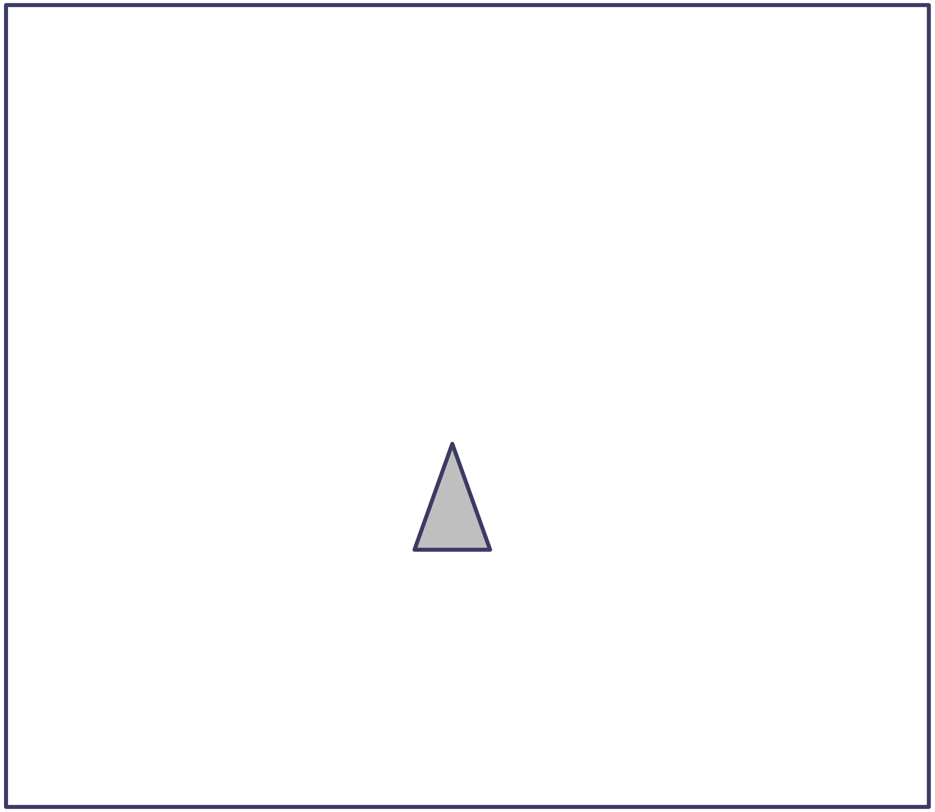


Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data,
distance data,
trees, networks)

Objective function (e.g.
parsimony, likelihood,
minimizing reticulation)

A **heuristic** will give you a solution, but conveys no information about how far the solution is from optimality



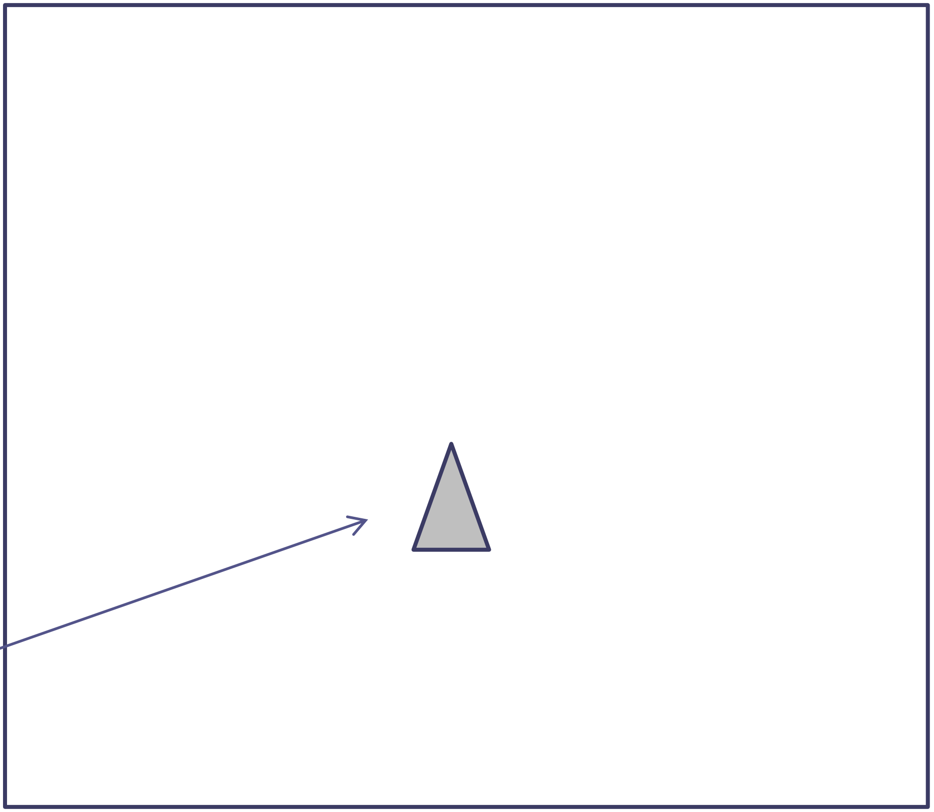
Space of feasible solutions (e.g.
space of trees, networks)

Input (e.g.character data,
distance data,
trees, networks)

Objective function (e.g.
parsimony, likelihood,
minimizing reticulation)

A **heuristic** will give you a solution, but conveys no information about how far the solution is from optimality

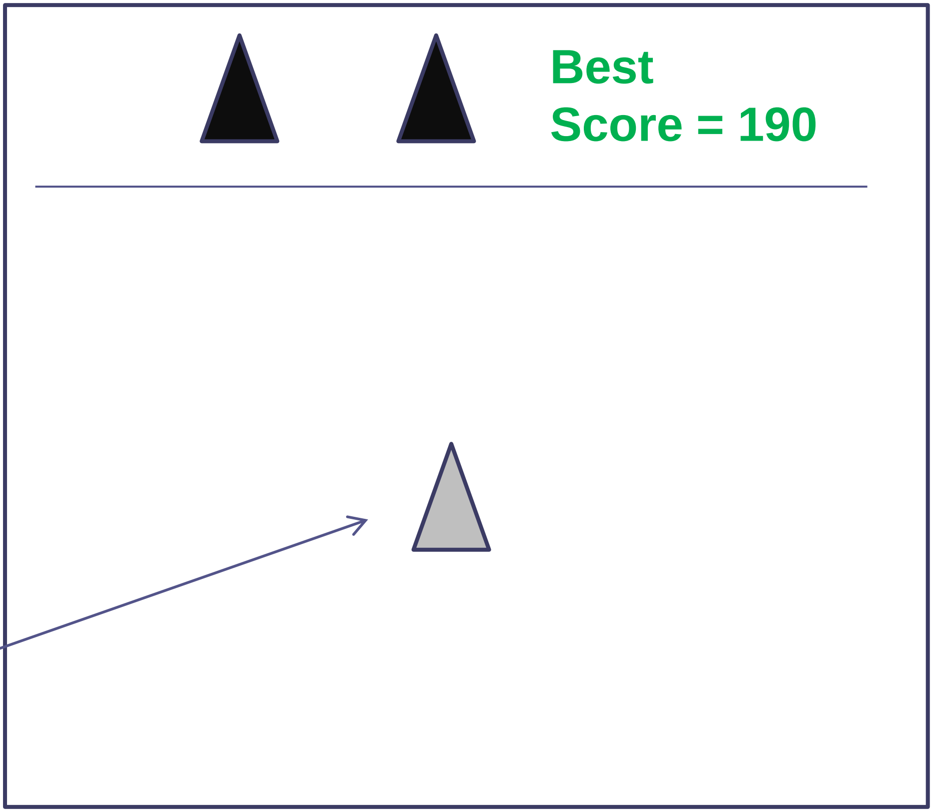
"I've found a solution.
Parsimony score 200!
Great!"



Space of feasible solutions (e.g.
space of trees, networks)

Input (e.g.character data,
distance data,
trees, networks)

Objective function (e.g.
parsimony, likelihood,
minimizing reticulation)



Best
Score = 190

A **heuristic** will give you a solution, but conveys no information about how far the solution is from optimality

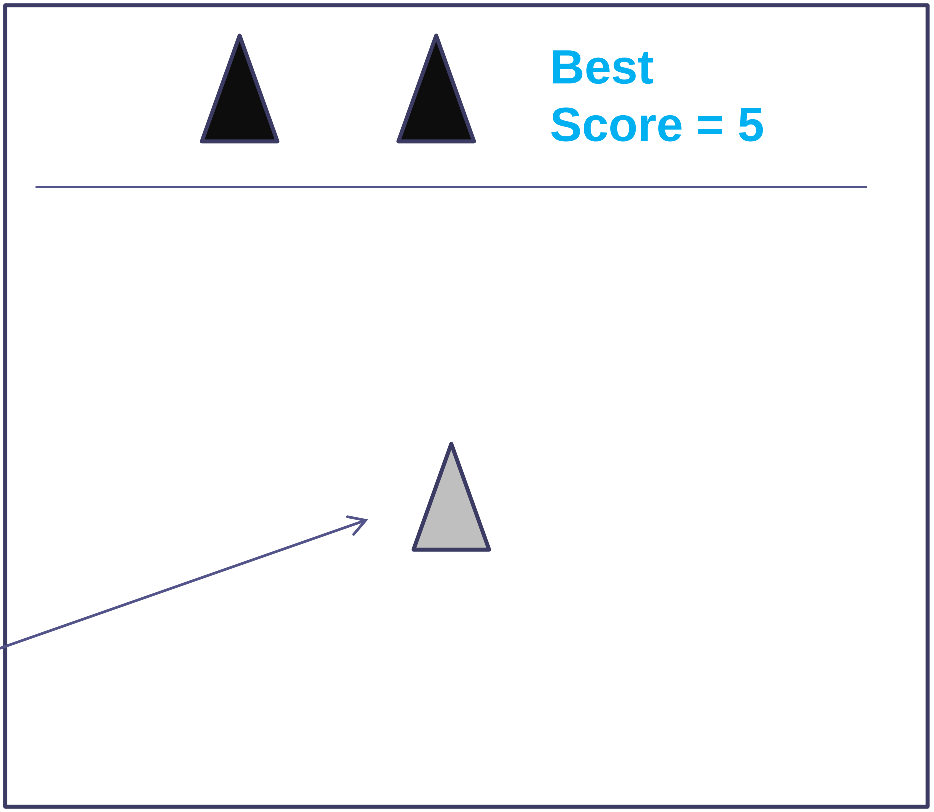
“I’ve found a solution.
Parsimony score 200!
Great!” **– indeed, not bad at all**

Space of feasible solutions (e.g.
space of trees, networks)

Input (e.g.character data,
distance data,
trees, networks)

Objective function (e.g.
parsimony, likelihood,
minimizing reticulation)

A **heuristic** will give you a solution, but conveys no information about how far the solution is from optimality



Best
Score = 5

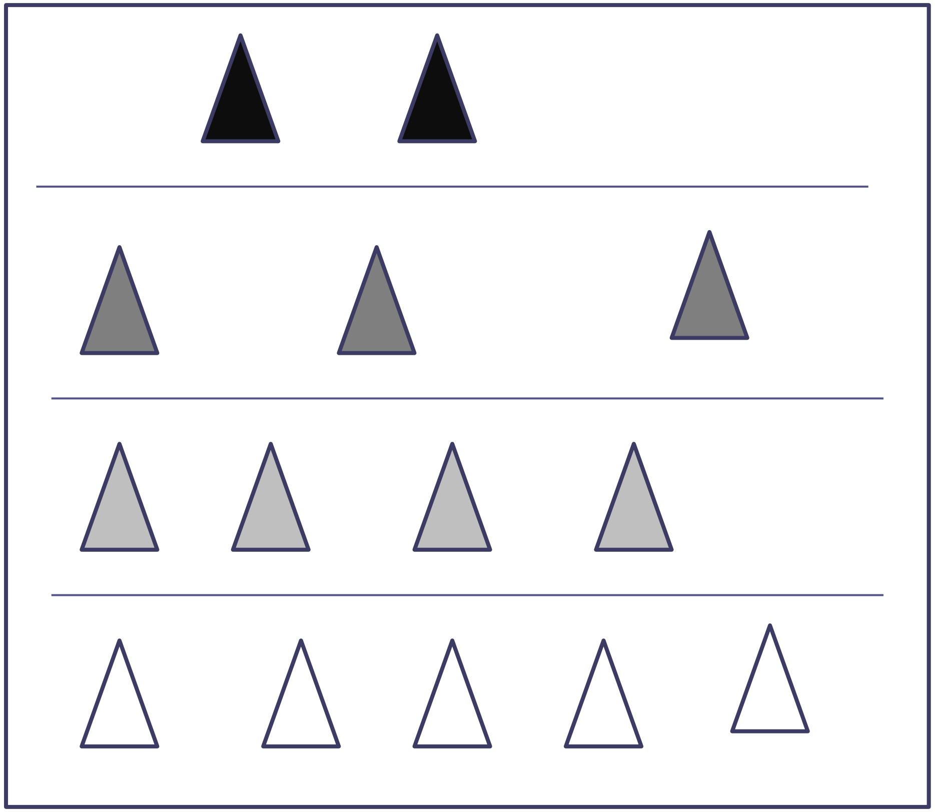
“I’ve found a solution.
Parsimony score 200!
Great!” **– no, really bad**

Space of feasible solutions (e.g.
space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

An **approximation algorithm** will give you a solution and some (pessimistic) measure of how far you are from optimality

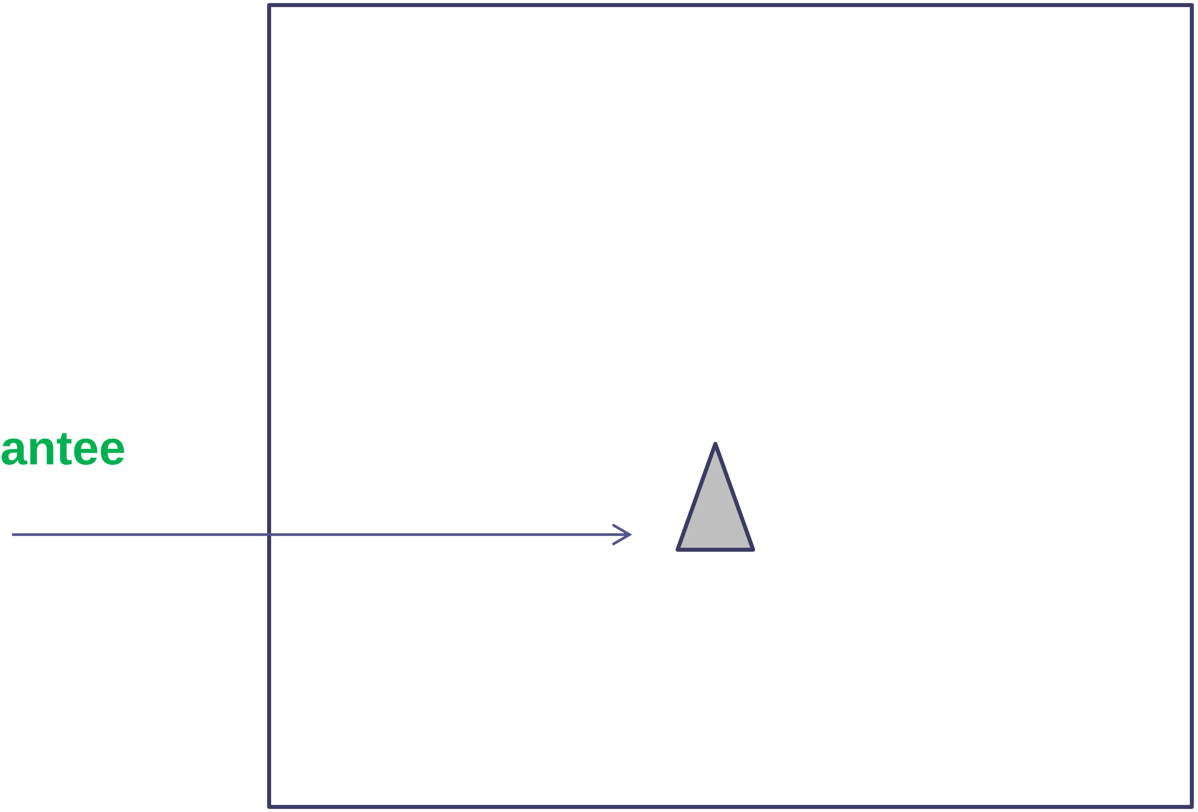


Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data, distance data, trees, networks)

Objective function (e.g. parsimony, likelihood, minimizing reticulation)

"I've found a solution. Parsimony score 200. Furthermore I can **guarantee** that it is within 20% of optimality."

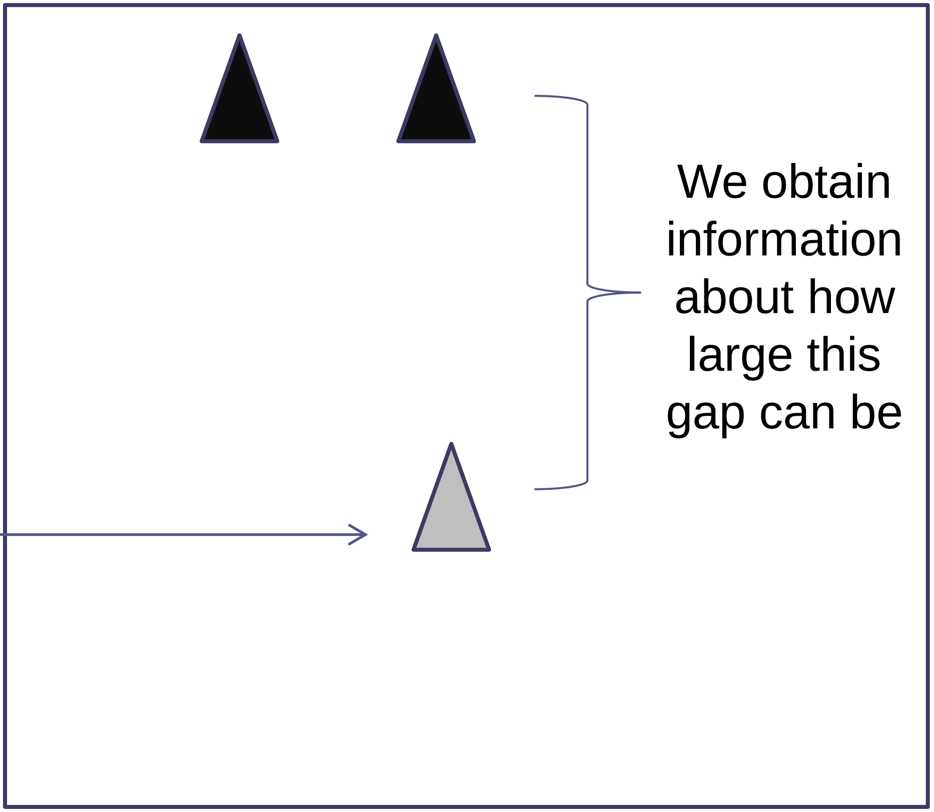


Space of feasible solutions (e.g. space of trees, networks)

Input (e.g.character data,
distance data,
trees, networks)

Objective function (e.g.
parsimony, likelihood,
minimizing reticulation)

"I've found a solution.
Parsimony score 200.
Furthermore I can **guarantee**
that it is within 20% of
optimality."



We obtain
information
about how
large this
gap can be

Space of feasible solutions (e.g.
space of trees, networks)

“Intractability” - suboptimality

- The obvious question is then: is it a good solution? This is the fundamental difference between **heuristics** and **approximation algorithms**.
- Certainly in my field, “heuristic” is a very dirty word, because they give us no reliable quality guarantees. So how do we know that the solution is “good”?
- There is of course still a role for heuristics, but for them to have any meaning they have to be **experimentally/empirically validated**. When this happens, they can be (much) more useful in practice than approximation algorithms – which are anyway difficult to develop.

Conclusions

- Mathematical optimization models can be a useful tool whenever biological questions can be framed in terms of “optimality”.
- The necessarily simplistic and restrictive character of such models is also a strength: it forces us to “codify” and “rank” implicit, qualitative and complex knowledge.
- Be careful not to over-interpret the output of such models, especially when they have not been experimentally validated.
- Arguably, the safest approach is to use mathematical optimization as an instrument in a more conventional biological analysis.
- Computational intractability is a problem that threatens our ability to compute optimal solutions, and inevitably influences the modelling process. However, many techniques are available for dealing with “intractable” problems.
- Be wary of any software package which cannot explain what exactly it is optimizing, or the proximity of its solutions to optimality!

Thanks for listening