A Crash Course in Gene and Genome Annotation

Lieven Sterck,
Bioinformatics & Systems Biology
VIB-UGent

Lieven.sterck@psb.vib-ugent.be

ProCoGen Dissemination Workshop, Riga, 5 nov 2013
"Conifer sequencing: basic concepts in conifer genomics"

"This Project is financially supported by the European Commission under the 7th Framework Programme"

Genome annotation:
finding
the biological relevant features
on a raw genomic sequence
(in a high throughput manner)

Thx to: BSB - annotation team

- Lieven Sterck (Ectocarpus, higher plants, conifers, …)
- Yao-cheng Lin (Fungi, conifers, …)
- Stephane Rombauts (green alga, mites, …)
- Bram Verhelst (green algae)

- Pierre Rouzé
- Yves Van de Peer

Annotation experience

- Plant genomes : *A.thaliana* & relatives (e.g. *A.lyrata*), Poplar, *Physcomitrella patens*, *Medicago*, Tomato, *Vitis*, Apple, *Eucalyptus*, *Zostera*, Spruce, Oak, Orchids …
- Fungal genomes: *Laccaria bicolor*, *Melampsora laricis-populina*, *Heterobasidion*, other basidiomycetes, *Glomus intraradices*, *Pichia pastoris*, *Geotrichum Candidum*, *Candida* ...
- Algal genomes: *Ostreococcus spp*, *Micromonas*, *Bathycoccus*, *Phaeodactylum* (and other diatoms), *E.hux*, *Ectocarpus*, *Amoebophrya* …
- Animal genomes: *Tetranychus urticae*, *Brevipalpus spp* (mites), ...

Why genome annotation?

- Raw sequence data is not useful for most biologists
- To be meaningful to them it has to be converted into biological significant knowledge : markers, genes, mRNAs, protein sequences
- Annotation is the first step toward this knowledge acquisition

Gene prediction

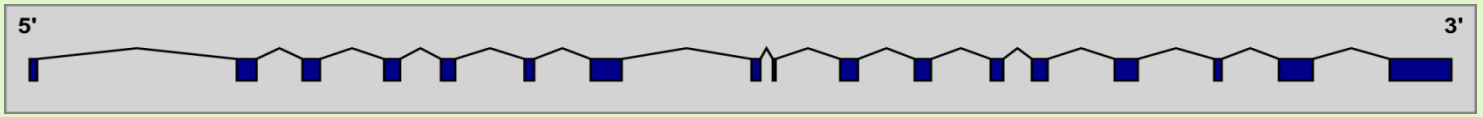
The process of
eukaryotic gene prediction
is well understood,
with many successful algorithms
using different approaches, though…

→ **Is by no means a solved problem**

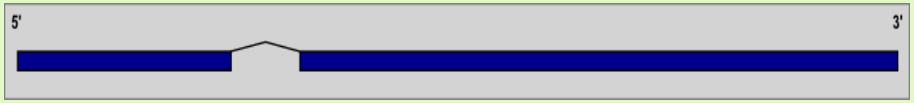
Gene Structures Differ

Different organisms can have different gene structure for the 'same' gene

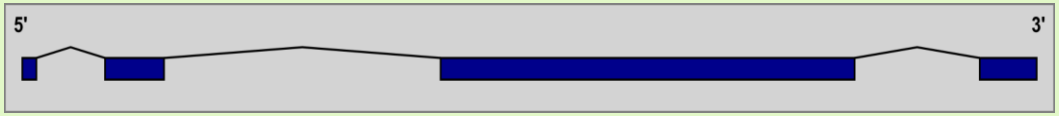
Ectocarpus



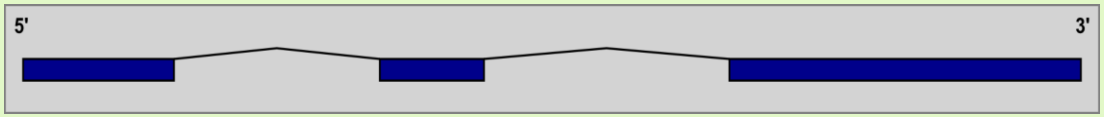
Ostreococcus



Populus



Spruce



Unsolved problems…

- New genomes require new approaches or at least require re-optimilisation of software
- Isoforms (NGS data)
- “New genes” , eg. lncRNAs, anti-sense RNAs, …
- New “organisation” of genes
  - Overlapping genes (transcript, CDS)
  - Nested genes
    - Genes in introns
    - Intermingled genes on opposite strands

How to do annotation?

- **Automated**: computationally generated via algorithms, highly dependent on transitive events
  - Fast, cheap, accuracy can be compromised, rapidly updated

- **Manual**: a trained human "annotator" views the annotation data types and interprets the data
  - slow, expensive, accurate, slowly updated

Automated <-> Manual

- Due to the VOLUMES of genome data today, most genome projects are annotated primarily using automated methods with limited manual annotation

- €€€€ and interest determine the blend of automated versus manual annotation for a genome

Genome Annotation: 2 steps

- Structural annotation / gene prediction

  'Where are the genes?'

  Defining **gene models** in a genomic sequence

  – Exons - introns

- Functional annotation

  'What is the function of the encoded protein/RNA?'

  Defining a **biological function** to the gene models

  – classic genetic/molecular bench work

  – transpose from what is known (homology searches)

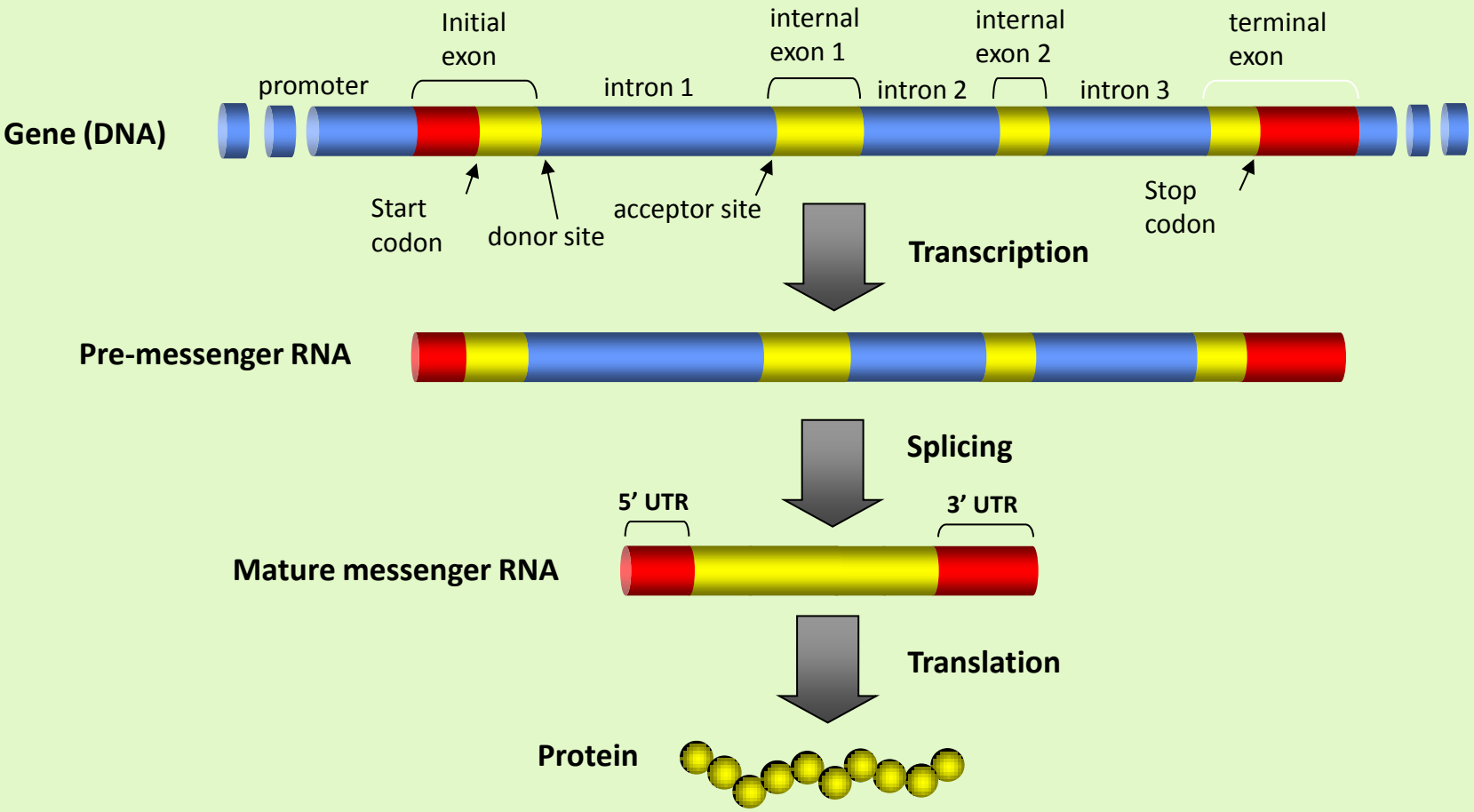
  – Expert input!

Structural annotation

Getting the gene structures from the raw sequence

ProCoGen Dissemination Workshop, Riga, 5 nov 2013
"Conifer sequencing: basic concepts in conifer genomics"

Basic Eukaryotic Gene Structure



AGATCAGATCCTTAGGATTCTGCAGGAGAGGCTTGTGGTAAGTCTTAAAAGATGCCTTTAATGAGAGAGGAATG
GCTGATTAATGGTATTCAAGGAATTATGAAGAAAAAATGCATAGTAGATTAACATGTCAATAACAATTTTTGAC
CCATTTAAGGACCTTGAAACTATATGATGATATATCTAATCTAAACTTATGGTACTTCTATGTTGATCTTAAAT
TATTATTTTCACTATTTGTAACAGGCACTTCCATATAAAGCATTTCAGTCCAATGCCTTGGAAATATGTGCAAG
AGTAAGAAAATATATGTCATTTTTTTATTTCTCTCAACCTTGTTTTCATTTTGATATTCTTTAATTGATATTCA
TAATGTTTGCATTGTTCCAGAAAGTATCAGCTGCATCAGGAGACATGAGAAAGGCTCTATGTGTCTGCAGGTGA
AATCTCTTACGAATTTCGTTTCCTTTGAGTTTGTTCTTGTATTGCATATGATGAATGAGATAGAAATGGGCTTT
AGAAACATGTCGTCGGTTCTAGATCGGAGAGTATCGGAGGTGTTAGATCCGCCGAAGTCAACACTTCTCGGAAA
CGGAAGTTGATATCCGATTCGGCTGCGGAGGTTTCAGCGACGGTGGTGTTGCCTGTGAATTCGATATCTACACC
GATGAAATGGAAATCGCCTCGTCGATGTGCTGTTTCAATCCCTAAGACCTCTGATGAGGTAAGCTCCTTGTTCT
TAGTTTAATTGTTTGGTTTAGTTTATTCGAGTGAGAGAATCAATTTGATTGTAGTTCTGATTGCAGGAGATTAA
GGAAGATTCTAATGAAAAATTGGAAAATCCAGTGATATCTGTTTGTTTAGAGGTTAAATCAAAGTGGAACCCTA
AAGGTATGTATATATGATGAGCTTTTTGATTCTCATGCTCTTAGTTAGTTTGATTGTATGATTGATGTATCTTG
AATGGATACAGATGATGAACAAATGAAAGCTGTGAAGGAGGCATTACATGTATCTAAGGCACCGTCAACTGTTG
TTTGTCGTGAGGATGAGCAAAGACGGGTTTTCGAGTTTGTTAAAGGTTGTATGGAGCAGAAGAAGGCTGGTAGT
TTGTATATATGTGGGTGTCCTGGAACTGGAAAGTCACTATCCATGGAGAAAGTAAGACTACAAGCTGAAGAATG
GGCAAAGCAGGTAACCAATCTTTGTGCATGTTTCATAAATTGTGATTAGTTTGAATCTTTCAATAGAAGGATTG
ATCTTTCATGAATTCTTACAGGCAGGTTTGCATTGTCCTGAAACAGTGTCTGTTAATTGCACATCACTGACAAA
AAGTACAGATATTTTCTCCAAGGTAAGACCTTATCCAGTCGAAAATTACTCGAAGTTATTGGTCTCATCATTTC
AGTATAGTTGTTAATTGAGTTATTTTGTTTTTAGATACTTGGTAACTATGAGTCCGGGAAGAAAGCTAATGGTT
CATTTTCACCTCTACAACAACTTCAGAAGATTGTTTTCTCAAAAGCAACAACAATCCAGATCAAAGATGATGTAG
GTTTTTCTTTTGTTTTTGGTCTTGTGATATGATGACCTGCTAGATAAGTAGATATAATTTGTTGCGCTGTTTCT
CATTCAAACTTTATCTGCAGGCTAATAATTGCAGATGAAATGGATTACTTGATCACAAGAGACCGAGGTGTCCT
TCATGAGCTTTTTATGCTCACAACTTTGCCATTGTCAAGGTGTATACTTATAGGTACTGTCTTTTGCGTTATTA
ATGTTCATTTTTTAAAATCTGTTTCCTATGGTCAAACAAGTTTTAAGTTTAAAGTTCGTATTTGTCCTCCAGGT
GTAGCAAATGCAATAGACCTTGCAGATCGTTTCCTTCCAAAATTGAAGTCTCTTAATTGTAAGAATTTCTCCAT
CACTGTGTGGTGCCATTTGAACACTCTTCTTTCTTCACCATATGTACCTAAGCTTTGATGAAGATTAAAAGTTG
AGTATACTTCTGTAACTTATACTTCCATCCAATGCAGGCAAACCTTTGGTTGTCACTTTCCGTGCCTATTCTAA
ATACTGTAAATTCACTCTTAAAATCATTACATTAAAGATGACAATCTCAGTTAACTCACATTTTTTTTCAATTT
CCAGGAGCGCTCTAGAAATCTTAGAAATAGAAGTTAGAGGATCAATAGATCAAGAACCAAAAGGTCCAGTTCCA
GAGTGTCAAGTGGTTAGTATTTTACTCTCTACTTTACTCAGTTCACTTACTGACGATGCA

Structural annotation

– Experimental (ESTs, cDNAs, RNA-seq)

Isolate and clone cognate transcripts (as cDNA) sequence them and compare cDNA with genomic DNA

It's the ONLY secure method!

BUT !!!

- Cloning is time consuming.
- Lowly expressed genes are difficult to detect.
- The nucleotide sequence does not contain translation information.
- Short read length of most RNA seq data

– Predictive

- intrinsic / *ab initio*
- extrinsic / comparative

Structural annotation

- Predictive methods
  - Intrinsic approach
    - Use of information/features from the sequence itself
    - Coding sequences have different properties than non-coding sequences
      - Codon-usage, GC-content
    - Requires training !!

Training datasets

- build training- and datasets of certified gene models (transcript alignments)
- Datasets are used to build HMM and IMMs; is mostly a collection of base sequences that are coding/non-coding
- Training sets are used to train start site, splice site and other intrinsic prediction tools

Intrinsic (*ab-initio*) approach

What kind of features?

Signals: TATA box, promoters, *cis*-acting motifs, transcription start sites (TSS), splice sites, polyA sites, transcription termination sites (TTS)

Contents: codon usage, GC content, nucleotide composition, base occurrence periodicity and hexamer frequency.

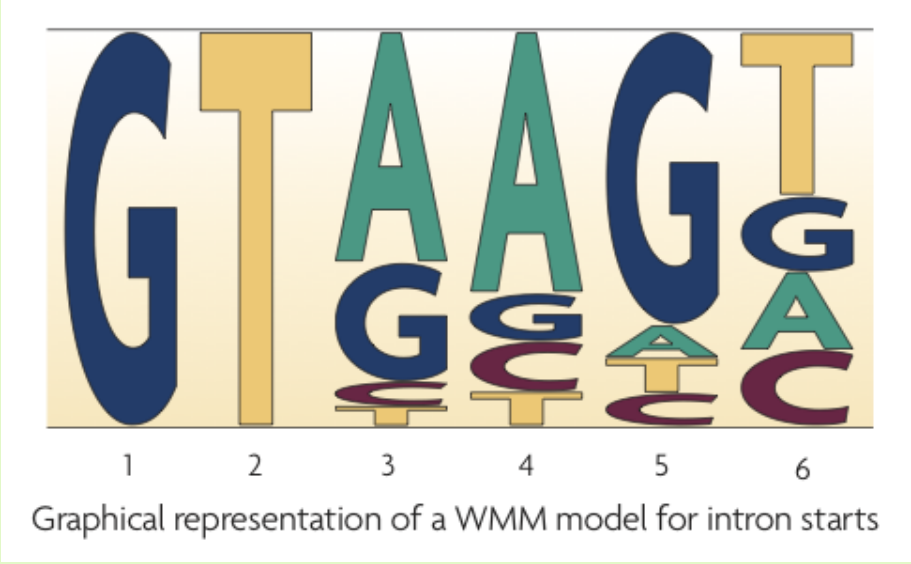
Method

Using a probabilistic models to predict features.

Signal sensors, simple example

- Weight Matrix Model (WMM)
- calculate the probability of *n*-mers in a given position

GTAAGT 9x
GTGGAG 2x
GTGTTA 5x



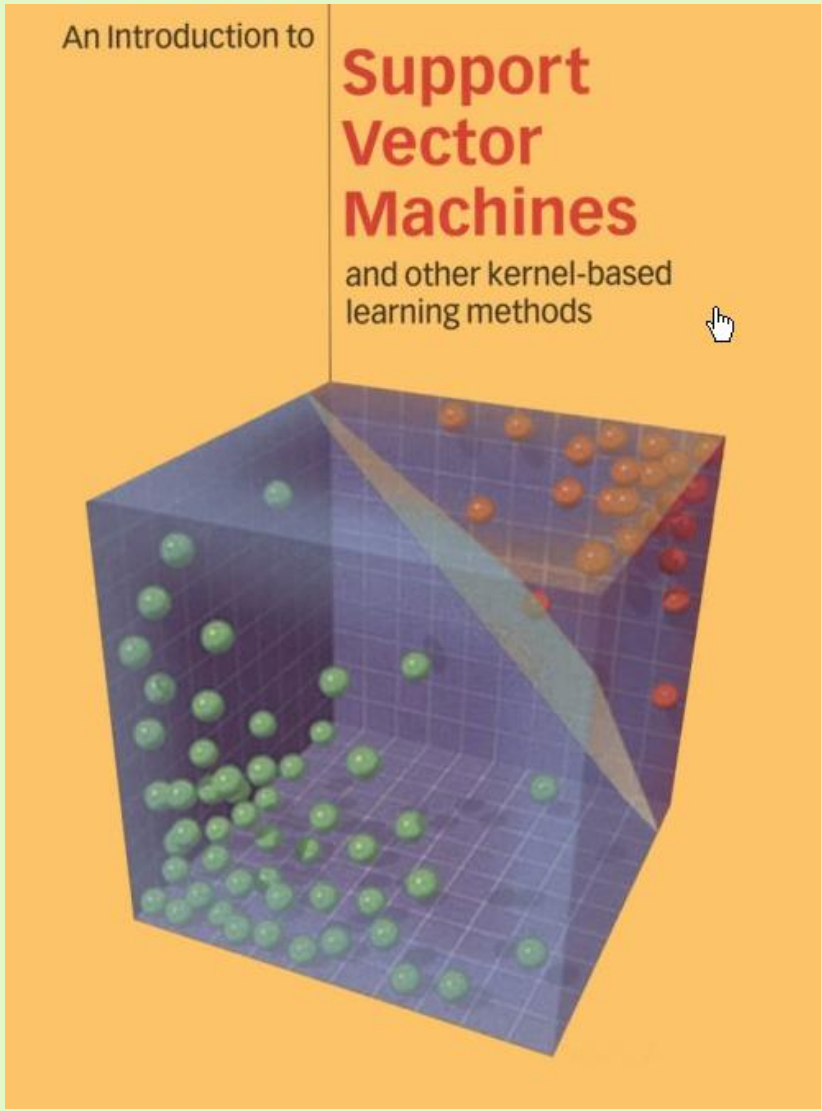
Graphical representation of a WMM model for intron starts

Can we expect to see GTAGGT as donor site?

Signal Sensor, advanced

- SVM modelling
- Each site is represented as a feature vector that is a point in an n-dimensional space.

- We employ a large margin hyperplane induction method (Suport Vector Machine) to build a decision function in this space.



Content Sensors

- Markov models (HMM & IMM)
- Compute the probability that a nucleotide sequence is coding/non-coding.
- Use conditional probability of a sequence position given previous *k* positions in the sequence.
  - Eg. 8th order Markov model: use previous 8 positions to calculate the probability of the 9th position (makes use of frequency of X-mers)

  -> training dataset

Intrinsic approach: drawbacks

- Needs training sets of documented genes
- Is not universal: All eukaryotic genomes maybe use the same DNA-language but each organism uses it's own dialect…
  - Needs training for every genome (= different training sets)
  - the used algorithms don't perform equally well on every genome
- Used algorithms are far from flawless

Structural annotation

- Predictive methods
  - Extrinsic approach
    - Search for similarities in protein and nucleic acid sequence databases

      (many genes are already documented, maybe there's an homologue on the genomic sequence)
    - Genomic DNA comparison between closely related species (eg. Human-Mouse)
    - Transcript alignments

Extrinsic approach: drawbacks

- Protein information
  - Need for homologues sequences
  - Species-specific genes are not detected, nor are fast evolving genes
  - Homologues sequences have to be full length and correctly annotated
  - Difficult to determinate correct exon-intron boundaries
- Nucleic acid information
  - cDNAs are often incomplete
  - Poor quality of EST sequences
  - Unequal coverage of the genes: dependant of the gene expression level

RNA seq issues

- RNA seq reads are often very noisy, with reads aligning allover the genome
- RNA seq detect sense and antisense transcripts, protein-encoding or not
- Only stranded reads can tell for sure in which sense they should apply
- Small read length of RNA seq leads to virtual transcripts, but are they (all) real ones?
- Artificial merging of (overlapping) transcripts

Extrinsic approach: drawbacks

- The 70% hurdle:
  - on average, this approach supports ~30-40% complete gene information and another ~30-40% genes are supported with incomplete information.
  - Transcript libraries only contains about ~60 -70% of all transcripts

=> despite NGS techniques this still, to some extent, holds true

Gene finding approaches

Gene finding programs can be classified into several types:

- **(1) ab-initio, ad hoc.** These apply an ad hoc scoring function to the set of all ORFs and then predict only those ORFs scoring above a predefined threshold.
  Examples are Glimmer and GlimmerM.

- **(2) ab-initio, probabilistic.** These adopt a rigorous probabilistic model of sequence structure and choose the most probable parse according to that probabilistic model.
  Examples are GenScan and TigrScan.

Gene finding approaches

- **(3) similarity-based.** These utilize evidence in the form of homology. These can be either ad hoc (eg., Grail, GeneWise, Exonerate) or probabilistic (eg., TwinScan, Slam, Twain).

- **(4) combiners.** These combine multiple forms of evidence, such as the predictions of other gene finders, and use ad hoc methods to arrive at a consensus prediction.
  Examples: GLEAN, EVidenceModeler, Maker

- **(5) Integrators.** These integrate multiple forms of evidence and use probablistic models to influence ab-initio predictions
  Examples include Augustus, FGenesH and EuGène

Gene finding –Ab initio–

- There has been a long history of successful ab initio programs
  - Genscan (Burge and Karlin 1997)
  - GeneMark.hmm (Lukashin and Borodovsky 1998)
  - GRAIL (Xu et al. 1994)
  - GlimmerM (Pertea et al. 2002)
    - authors used the HMM framework to provide the parameterization and decoding of a probabilistic model of gene structure

      (for review, see Zhang 2002).

Gene finding –similarity based-

- Not much prediction, *sensu strictu*
- Restricted to what can be aligned
- Only possible when (closely) related gene and genome sequences exist
- or entirely dependent on EST (cDNA) or RNA-Seq
- Will not ‘predict’ fast evolving genes or ‘new’ genes
- No need for training set! Ready to be used

Why use protein similarity

- EST and cDNA collections and even RNAseq libraries are never exhaustive in any organism.
- Exploit the wealth of data available from other species, through protein homology.

- But to be treated with caution…
  - Error propagation
  - Preferably use only non-predicted proteins!
    - Swissprot, experimentally verified proteins, …

Combiner approach

- Do not have ab-initio prediction capabilities
- Combine several types of information to get to a consensus
- Are restricted to what the data types offer them
- Usually also input from ab-initio predictors

Integrative approach

The question we endeavor to answer is:

"What is the most probable collection of gene structures in a genomic sequence, given all of the available evidence?"
(using statistics generated from a training set)

Thus: use what ever is available

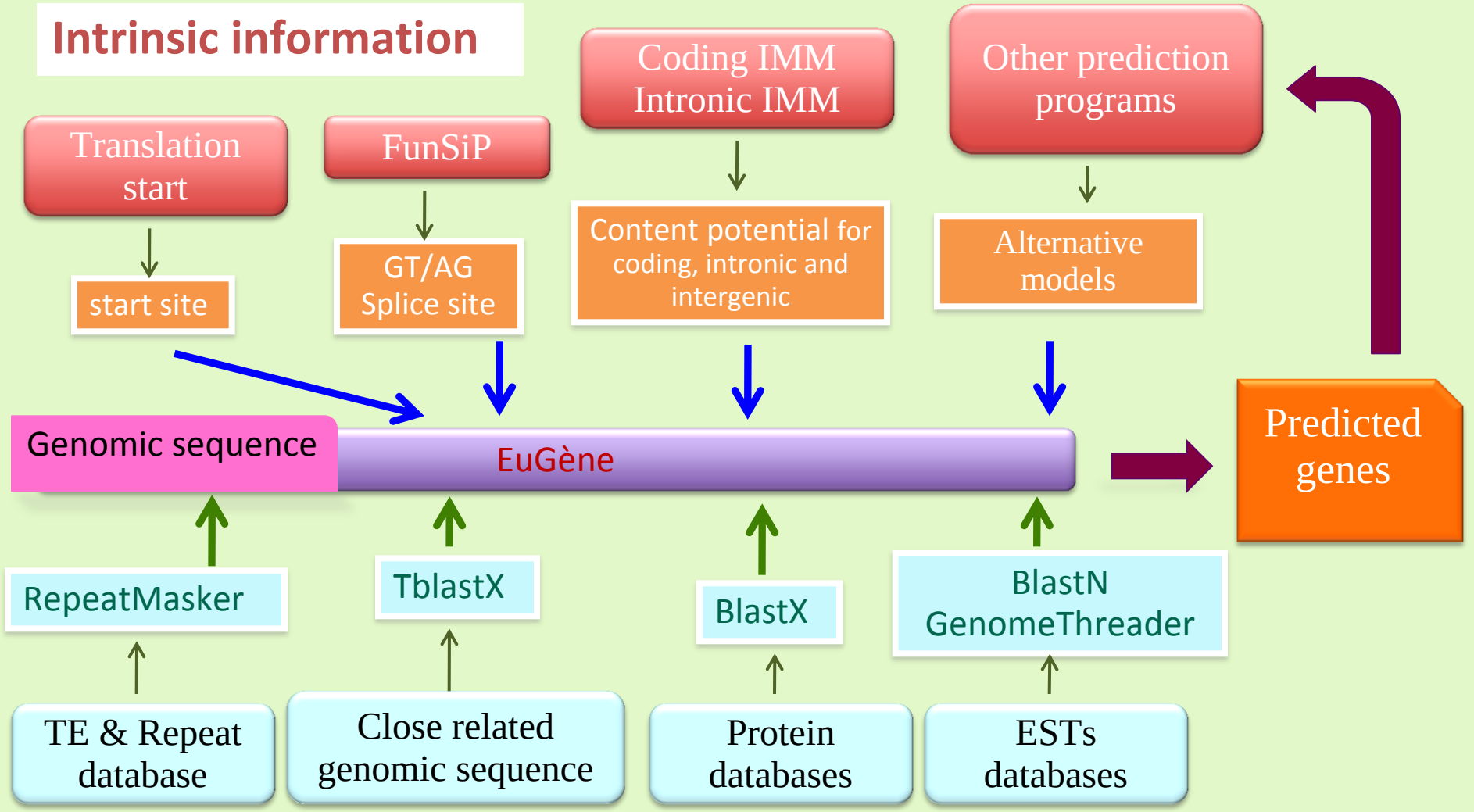
Gene finding –integrators-

- Integrative gene prediction programs are ones which use external evidence to influence the scoring of *ab initio* schemes,
  - *Ab initio* : thus requires training!
- *A posteriori* approach
  - Includes algorithm to correct *ab initio* prediction based on transcript and/or protein alignments
    - examples:
      - Genie (Kulp et al. 1996),
      - Fgenesh++ (Solovyev and Salamov 1997).
      - Genomescan (Yeh et al. 2001),

Gene finding –integrators-

- *A priori* approach
  - Algorithm is first informed about transcript and/or protein alignments. *Ab initio* part takes alignment information into account to design gene structures
    - examples:
      - EuGène (Schiex et al., 2001,2008)
      - Augustus (Stanke et al. 2006)

EuGène: gene prediction platform



EuGène – Motivations

- A thorough annotations must take into account:
  - similarities with known sequences (proteins,EST/cDNA...)
  - region content analysis, lengths
  - signal prediction software (ATG, splice sites. . .)
  - integrated prediction tools (GenScan, GeneID. .)
  - all available significant biological knowledge.

- This a slow painful manual process
  Try to automate this as far as possible.

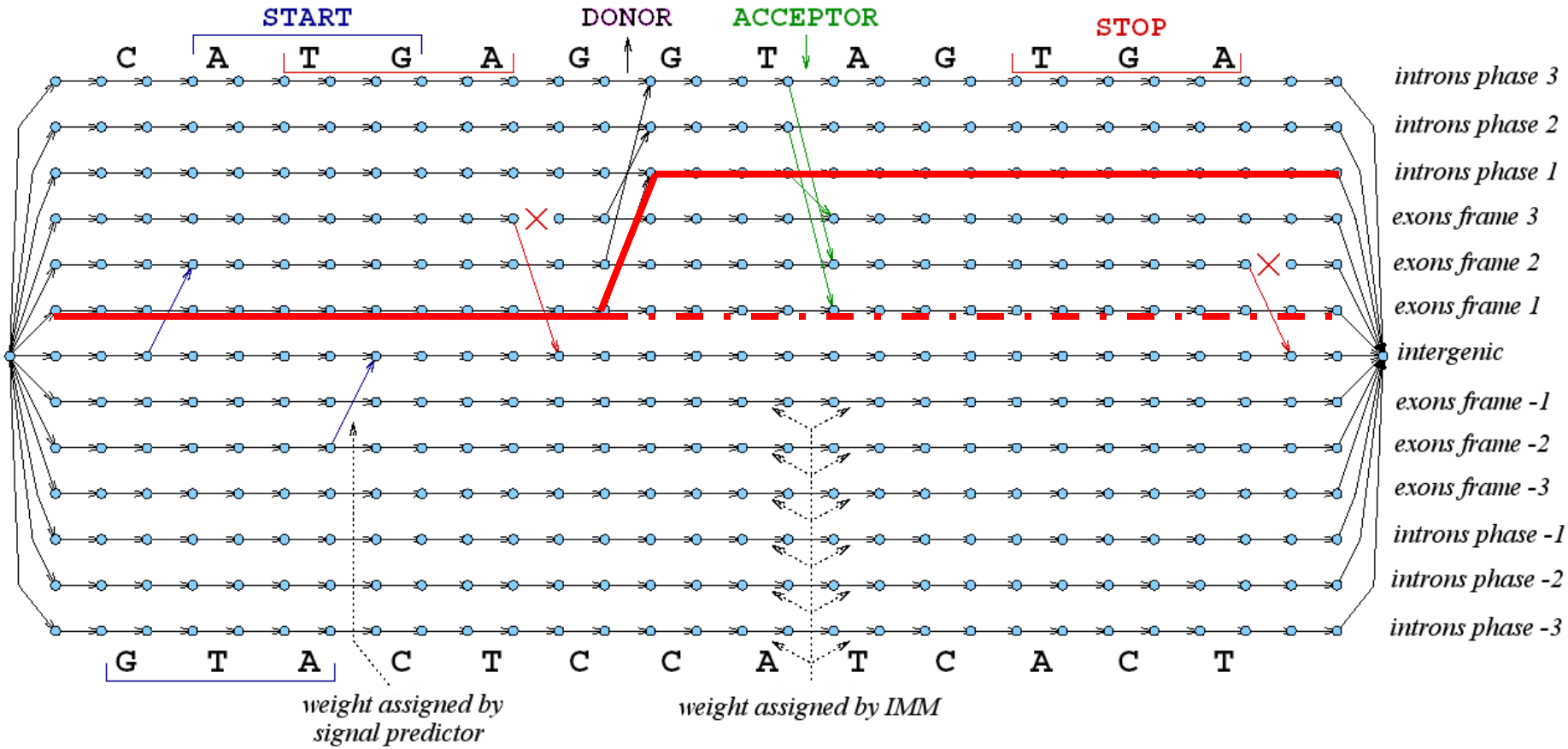
EuGène

- A priori or informed approach
  - *ab initio* part 'knows' about available local extrinsic information
    (alignments of EST, cDNA, proteins)
  - Prediction is the shortest path trough the model exploiting the data to its best performance
  - Need to train all the models for *ab initio* and weighting of all extrinsic data

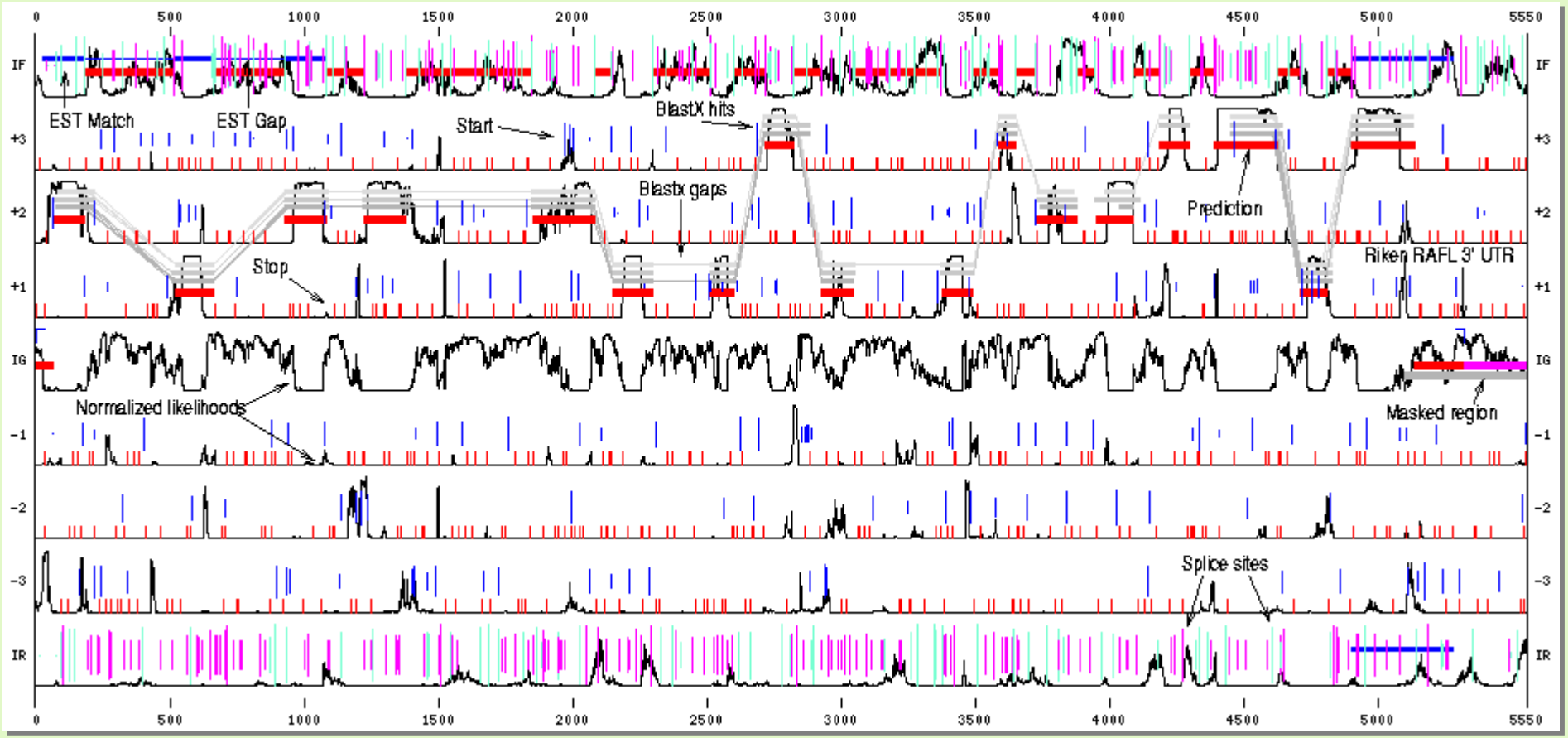
EuGène prediction graph

- Directed Acyclic Graph
- each base is represented individually (a dot)
- 2 adjacent bases may be separated by « signals » (=an edge)
- a path in the graph = a consistent prediction
- weighting, removal, addition of edges according to available evidence.

EuGène - DAG

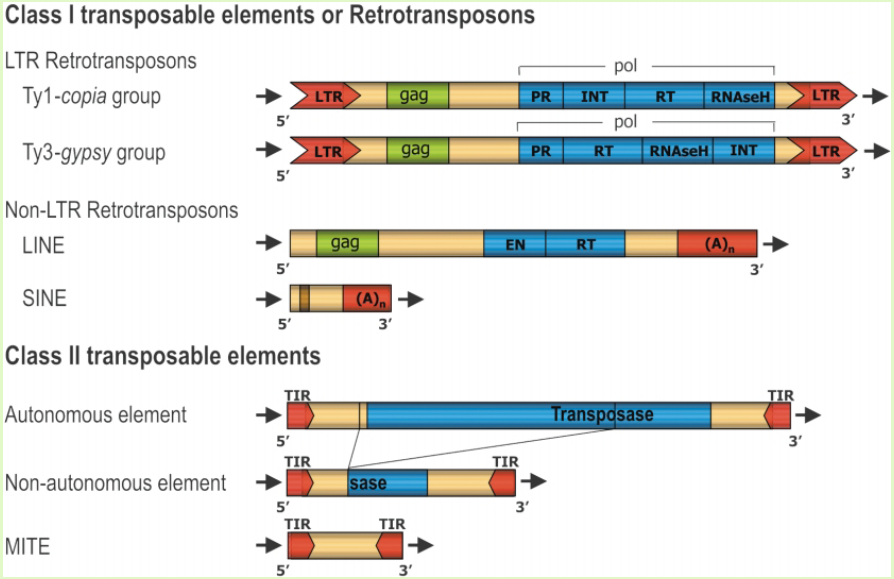

START
DONOR
ACCEPTOR
STOP
C A T G A G G T A G T G A
introns phase 3
introns phase 2
introns phase 1
exons frame 3
exons frame 2
exons frame 1
intergenic
exons frame -1
exons frame -2
exons frame -3
introns phase -1
introns phase -2
introns phase -3
G T A C T C C A T C A C T
weight assigned by signal predictor
weight assigned by IMM

Output of Eugène



Masking transposons

- Build database of transposons (TE)
  - Know transposons
    - Seek for a (better) homolog in the genome
  - Search for species specific TE
- tools
  - TESeeker
  - RepeatMasker
  - RepeatModeler

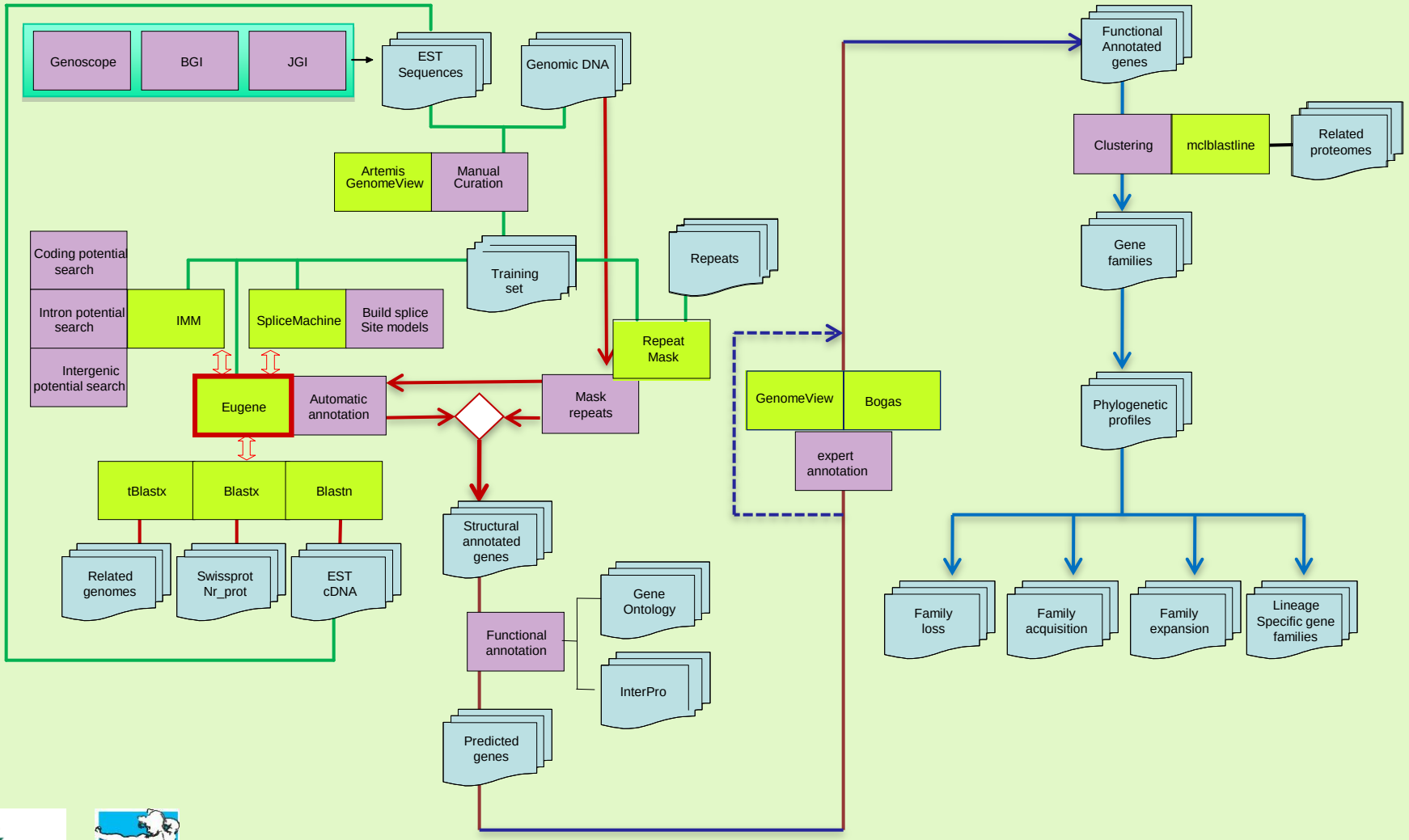


Not as straightforward as it seems, some proteins in the ATH & rice genome could still probably be candidate TE

Why repeat masking

- TE have their own protein encoding genes, so although they are not proper genes from the organism, they still show, up to some degree, the same characteristics as real protein-coding genes.
- Most gene calling tools confuse then TE with real protein encoding genes, which then results in false positive genes or in the fusion with correct gene models

Our genome annotation pipeline …



We found the genes

What now?

ProCoGen Dissemination Workshop, Riga, 5 nov 2013
"Conifer sequencing: basic concepts in conifer genomics"

Gene prediction is not perfect

- Most gene-callers have a sensitivity and specificity roughly of ~60% to ~90% depending of the available extrinsic data, genome, program used, level of specific training …

→ so up to 40% of genes can have some kind of error ranging from minor things to completely missed genes

What can go wrong…

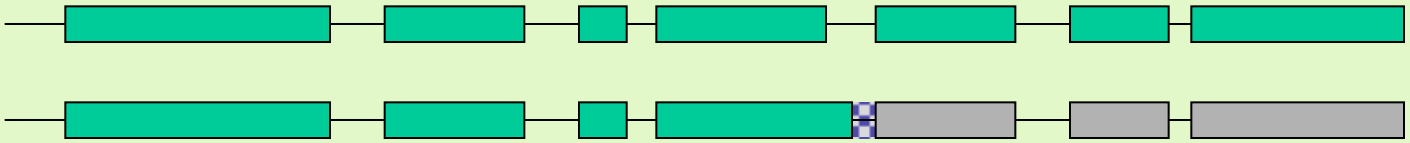
- Structural annotation
  - Signal peptides carry information in the sense of charges, chemical properties of the amino-acids,
    NOT based on a proper codon usage
    - start of protein missed, truncated sequence
  - Rare splice sites
    - Not enough data to build models
    - So rare that the cost to predict those correctly will be to high compared the gain in number of genes
      - To be corrected manually

What can go wrong...

- Splitted genes

  - intron longer than normal
  - weak splice site signal
  - other

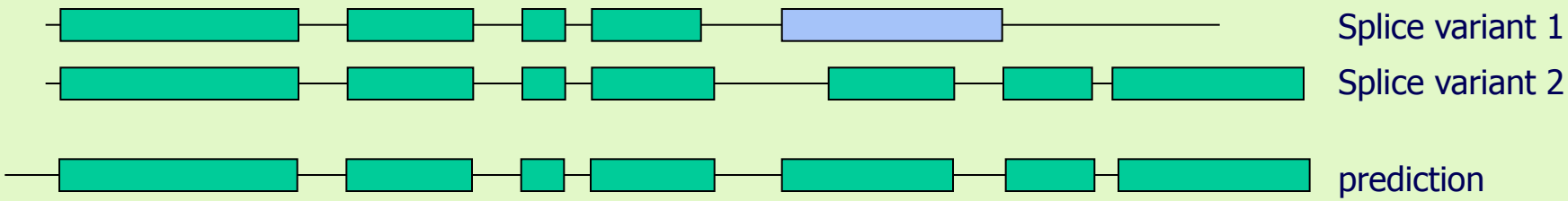
- Fused genes



  - intergenic region shorter than normal
  - weak splice sites
  - other

What can go wrong…

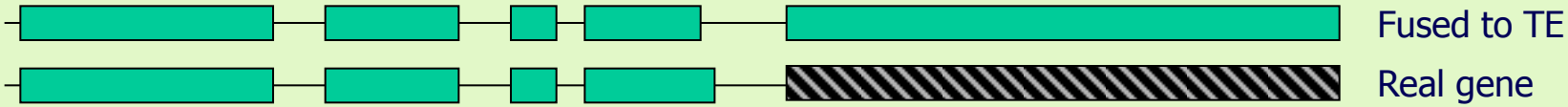
- Alternative splicing



- Predicted model is mixture of 2 or more variants or one of the variant (often the longest one)
- Hard to define/predict alternative splice-variants as both are valid models,
- programs often go for the longest one…
- **Only valid way to predict alternative splice-variants is through aligned ESTs and/or full length cDNAs (and preferably with several copies for each variant)**

What can go wrong...

- Fusion with transposable elements



  - Transposable elements(TE) have protein encoding genes
  - Those TE-genes confuse gene-callers

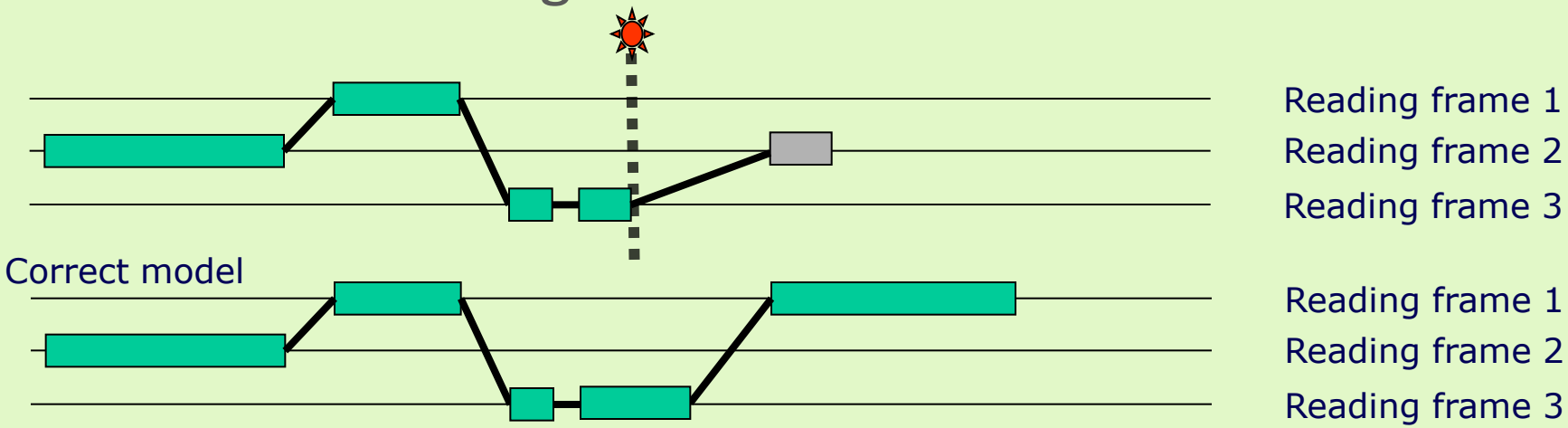
- Masking out of true genes
  - Highly conserved genes can be picked up by repeat masking tools, especially when they are very abundant

What can go wrong…

- Other issues:
  - New type of genes (formerly unknown)
    - Unknown: so models not appropriate
  - Too small genes, genes that are deviating from the general concepts… are those true or false positives?
  - Pseudogenes (conifers!)
  - Poor quality/state of the genome sequence
    - Fragmented assembly (eg. Spruce)

What can go wrong...

- Indels inducing frame shifts



- Due to indels in the genomic sequence reading frames can shift resulting in truncated genes (most of the time).
- Gene-callers will try to solve the problem by prematurely terminate genes or introduce introns to span an introduced stop-codon

Errors… solve them

- Systematic errors:
  - Re-train the program…
    - Build better models, larger training sets
    - Better masking
    - Other… until good enough
- Occasional errors
  - Effort to correct them sometimes to high…

    manual!
  - Leave it to the specialists working on the genes

    =**YOU**

    report corrections to the community, the community will be grateful

We have the genes, and the structure seems ok …

But what does this gene do?

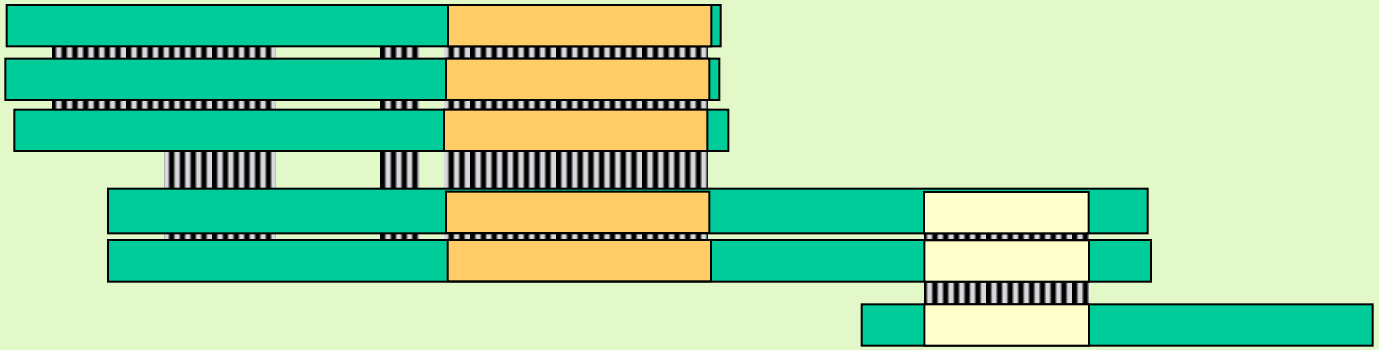
Functional Annotation

- Gene product names
- Gene symbols
- EC numbers
- Expression profile
- Gene Ontology assignments
- Other fundamental features of gene relevant to organism
- Text-mining

Functional annotation is mainly based on similarity to known proteins in AA databases (eg. SwissProt)

What can go wrong...

- Functional annotation
  - Relies almost solely on comparison with known sequences
  - Danger for error propagation!



Gene family A is homologue to gene family B

Gene family B is related to gene family C

But gene family C is unrelated to gene family A

Although in GenBank/EMBL/DDJ one can find functional annotation with descriptions for all genes as A-like

Typical for highly modular types of proteins

Functional Annotation: Gene Names

- Categories:
  - Known or Putative: Identical or strong similarity to documented gene(s) in Genbank or has high similarity to a Pfam domain; e.g. kinase, Rubisco
  - Expressed Gene: Only match is to an EST with an unknown function; thus have confirmation that the gene is expressed but still do not indicate what the gene does
  - Hypothetical: Predicted solely by gene prediction programs; no database match except possibly other hypotheticals (=> sometimes called conserved hypothetical protein)

We (automatically) determined a structure and a function

Job done?

Improving the Annotation

- Automated annotation is not bad, but manual annotation is better
- Problem for manual annotation is time consuming and goes “stale” quickly
- Thus, how does a community update the annotation
- Three models:
    - Don’t update annotation eg. MANY
    - Official annotation project (requires continual funding, restricted to large communities) eg. TAIR
    - Update through community efforts (highly focused, no mechanism to address whole genome, quality can be variable) => online resources, eg. ORCAE

Continuous Community Annotation

Online
Resource for
Community
Annotation of
Eukaryotes

Wiki-style platform to enabling efficient community curation of initial automated gene and genome annotations

13 public projects, 10 on-going projects

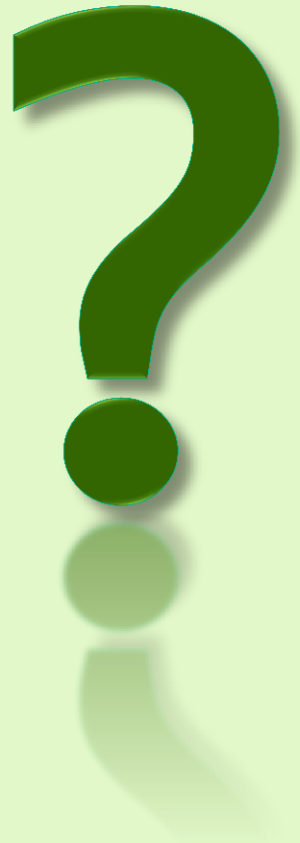
http://bioinformatics.psb.ugent.be/orcae/

L. Sterck et al. *Nature Methods*,9:1041 2012

Take home messages

- You can't go around annotation!
  - You'll need it! You'll use it all the time!
- Gene prediction is seldom perfect
  - Be aware of it and learn to deal with it
  - A quick check can save you a lot of trouble
  - Share your corrections
- We are all doing our best to deliver the best possible annotations…

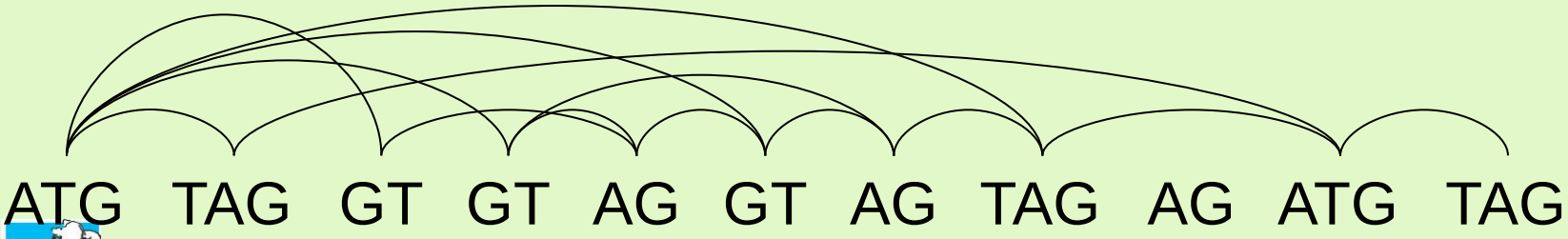
Thanks for your attention



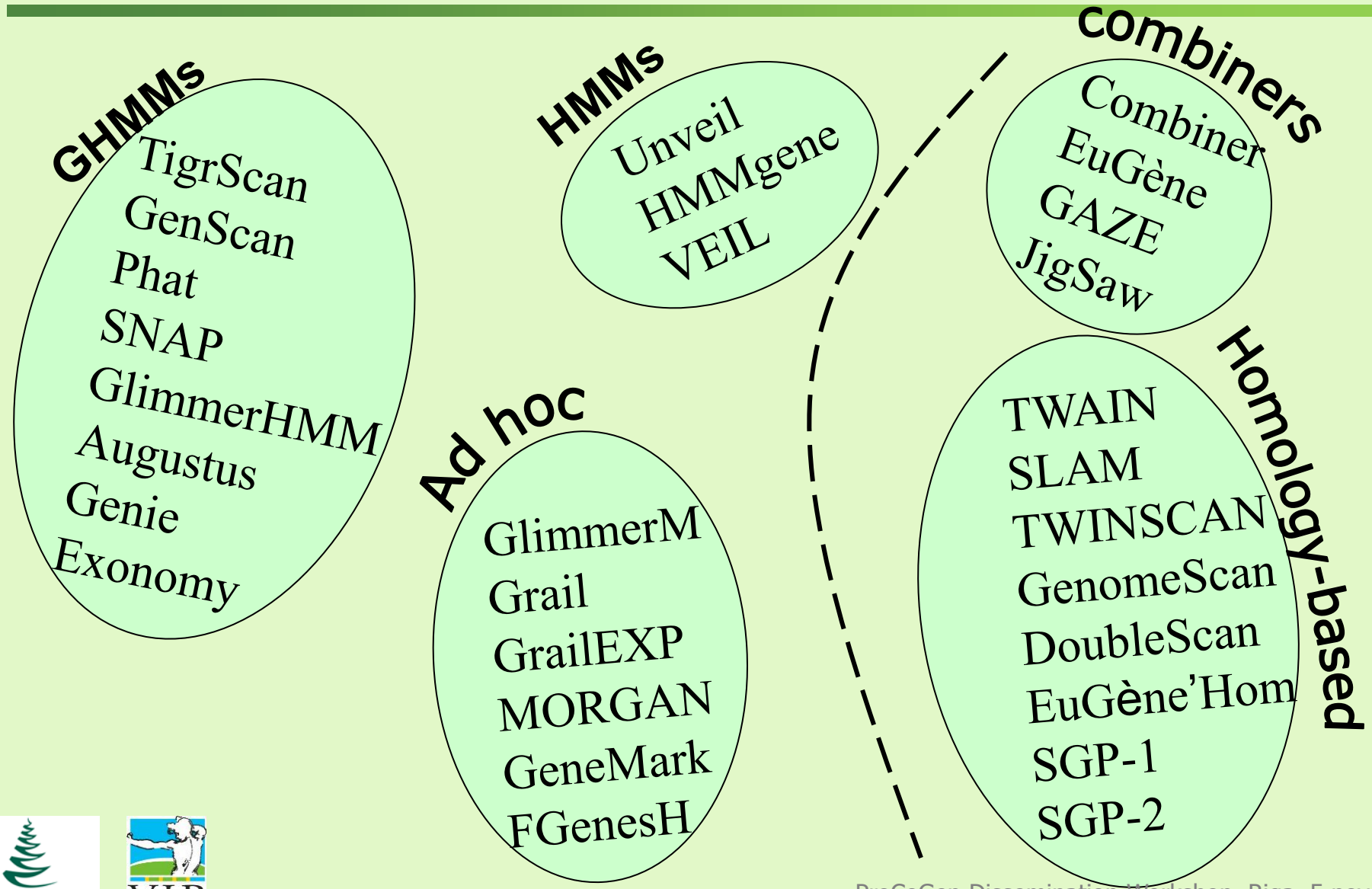
The (Eukaryotic) Gene Prediction Problem

- Gene prediction is the problem of parsing a sequence into non-overlapping coding segments (CDSs) consisting of exons separated by introns.
- Untranslated regions (UTR) are rarely predicted.

This parsing problem can be visualized as one of choosing the best path through the graph of all open reading frames



Types of methods



Gene finding –Alignment based-

- The development of pure evidenced-based gene prediction can be traced back to the 1996
  - Procrustes (Gelfand et al. 1996)
  - GeneWise (Birney et al., 2004) is a pair-HMM style method, with strong similarities to the more recent dual genome predictors
  - DoubleScan (Meyer and Durbin 2002)
  - SLAM (Alexandersson et al. 2003)
- pure feature-based programs, which have no inherent probabilistic model or knowledge of the underlying DNA, but provide a framework for the integration of component features which do have knowledge of the DNA sequence, such as
  - GAZE (Howe et al. 2002).
  - Genomewise (Birney et al., 2004)

Spliced Alignments

- Alignments taking into account correct splicing

```
Query+   ATGCAGCAGGTAAATTTTTGTCATCTTTTTTCAAAACTCCTGCAACAATAACCATAAAC
         ||||||||||----------------------------------------------
Squamo+  ATGCAGCAG

Query+   AAGAACAAGAAATATGAGACCTTTTACCTTTTGTTTTTAAGGTTTCACGCGCTCAGTGAG
          -----------------------------------------|||||||||||||||||||
Squamo+                                            GTTTCACGCGCTCAGTGAG
```

- Tools:
  - Transcript:
    - Classic: sim4, Genomethreader,
    - NGS: Bowtie/TopHat, GSNAP
  - Protein: GeneWise, Genomethreader

Combiners

Basic steps in structural genome annotation

(in theory, as done in Ghent)

- Mask the repeats and transposons in the genomic sequence
- Map cognate transcripts on the sequence (sim4, DNA-alignments)
- Map protein homologues (blastx)
- *Map other genomes (tblastx)
- Run content sensors and signal sensors
- Make prediction

Correct annotations –manually–

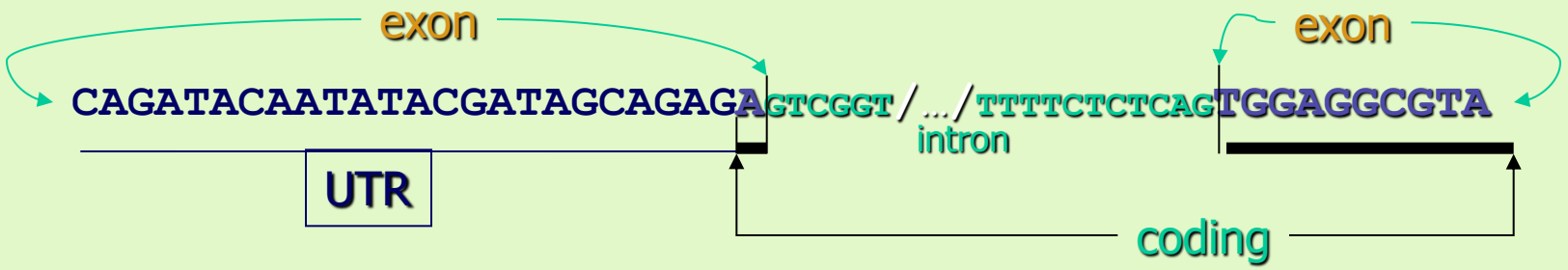
- Use as much as possible alignments to correct annotations
  - Transcript alignments are the most reliable
  - Protein alignments will confirm putative functional annotation

- Look at whole gene-families:

  → the best homologues are the paralogues

Correct annotations –manually–

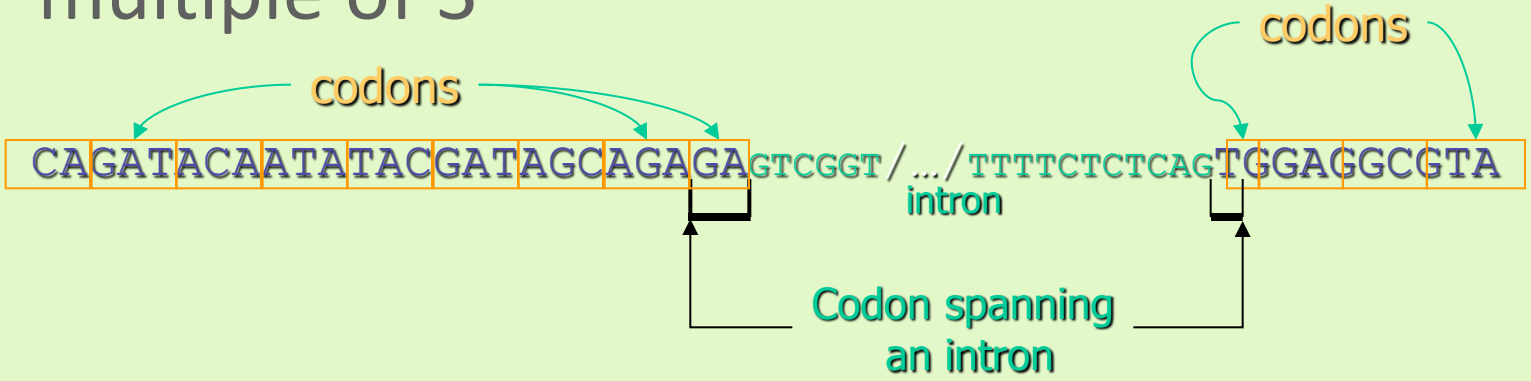
- Transcription, splicing and translation in eukaryotes are distinct processes



  - This means that CDS (coding sequences as reported in Genbank/EMBL files) can have an "exon" of size 1
    - True exon has UTR upstream of the CDS!
  - *ab initio g*ene-callers will be unable to predict those correctly need alignment to delineate exons correctly

Correct annotations –manually–

- Exons do not need to have a length equal to a multiple of 3



  - It is only after splicing is finished that the resulting open reading frame on the transcript has to be a multiple of 3
  - Take care for 'in frame stop-codons'

ORCAE