An Introduction to Conrad

Esfandiar Haghverdi

Indiana University Bloomington

April 16, 2008
Conrad: Gene prediction using conditional random fields

DeCaprio, Vinson, et al.
The Broad Institute of MIT and Harvard
*Genome Research, November 2007*

- Master’s thesis of Matthew Doherty.
Conrad: Gene prediction using conditional random fields

DeCaprio, Vinson, et al.
The Broad Institute of MIT and Harvard
*Genome Research, November 2007*

- Master’s thesis of Matthew Doherty.
- Conrad is freely available from
  www.broad.mit.edu/annotation/conrad
Conrad: Gene prediction using conditional random fields

DeCaprio, Vinson, et al.
The Broad Institute of MIT and Harvard
**Genome Research, November 2007**

- Master’s thesis of Matthew Doherty.
- Conrad is freely available from [www.broad.mit.edu/annotation/conrad](http://www.broad.mit.edu/annotation/conrad)
- More can be found there!
Conrad: Gene prediction using conditional random fields

DeCaprio, Vinson, et al.
The Broad Institute of MIT and Harvard

**Genome Research, November 2007**

- Master’s thesis of Matthew Doherty.
- Conrad is freely available from [www.broad.mit.edu/annotation/conrad](http://www.broad.mit.edu/annotation/conrad)
- More can be found there!
- Supplemental material at [www.genome.org](http://www.genome.org)
Key features of Conrad

- Based on semi-Markov conditional random fields.
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
- Pre-trained models for several organisms.
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
- Pre-trained models for several organisms.
- Easily trainable for single genome and comparative gene prediction.
- Ability to handle ESTs, BLAST, and other data types as features.
- Written in Java, runs on any platform.
- Flexible input and output data handling.
- Interface to define custom features and algorithms to extend the gene caller.
- Open source using the GPL license.
- Can be extended and applied to other problems!
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
- Pre-trained models for several organisms.
- Easily trainable for single genome and comparative gene prediction.
- Ability to handle ESTs, BLAST, and other data types as features.
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
- Pre-trained models for several organisms.
- Easily trainable for single genome and comparative gene prediction.
- Ability to handle ESTs, BLAST, and other data types as features.
- Written in Java, runs on any platform.
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
- Pre-trained models for several organisms.
- Easily trainable for single genome and comparative gene prediction.
- Ability to handle ESTs, BLAST, and other data types as features.
- Written in Java, runs on any platform.
- Flexible input and output data handling.
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
- Pre-trained models for several organisms.
- Easily trainable for single genome and comparative gene prediction.
- Ability to handle ESTs, BLAST, and other data types as features.
- Written in Java, runs on any platform.
- Flexible input and output data handling.
- Interface to define custom features and algorithms to extend the gene caller.
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
- Pre-trained models for several organisms.
- Easily trainable for single genome and comparative gene prediction.
- Ability to handle ESTs, BLAST, and other data types as features.
- Written in Java, runs on any platform.
- Flexible input and output data handling.
- Interface to define custom features and algorithms to extend the gene caller.
- Open source using the GPL license.

Can be extended and applied to other problems!
Key features of Conrad

- Based on semi-Markov conditional random fields.
- Better accuracy than HMM gene predictors.
- Pre-trained models for several organisms.
- Easily trainable for single genome and comparative gene prediction.
- Ability to handle ESTs, BLAST, and other data types as features.
- Written in Java, runs on any platform.
- Flexible input and output data handling.
- Interface to define custom features and algorithms to extend the gene caller.
- Open source using the GPL license.
- Can be extended and applied to other problems!

An Introduction to Conrad
The challenge

- Need for accurate and automated gene prediction methods.
The challenge

- Need for accurate and automated gene prediction methods.
- *Manual* curation is the best, but expensive and slow.
The challenge

- Need for accurate and automated gene prediction methods.
- **Manual** curation is the best, **but** expensive and slow.
- Best methods available are based on HMMs, GHMMs.
The challenge

- Need for accurate and automated gene prediction methods.
- **Manual** curation is the best, but expensive and slow.
- Best methods available are based on HMMs, GHMMs.
- External evidence cannot be readily incorporated, e.g. EST, BLAST, Protein database, etc.
The challenge

- Need for accurate and automated gene prediction methods.
- **Manual** curation is the best, **but** expensive and slow.
- Best methods available are based on HMMs, GHMMs.
- External evidence cannot be readily incorporated, e.g. EST, BLAST, Protein database, etc.
- A possible solution: annotation pipeline.
  - New organism application is labor-intensive and complex.
  - Uses heuristics.
The challenge

- Need for accurate and automated gene prediction methods.
- **Manual** curation is the best, but expensive and slow.
- Best methods available are based on HMMs, GHMMs.
- External evidence cannot be readily incorporated, e.g. EST, BLAST, Protein database, etc.
- A possible solution: annotation pipeline.
  - New organism application is labor-intensive and complex.
  - Uses heuristics.
- There are several theoretical extensions of GHMMs but not all encompassing.
Conditional Markov Random Fields (CRF) replace GHMMs.

- Incorporate evidence that contains unknown dependencies.
Conditional Markov Random Fields (CRF) replace GHMMs.

- Incorporate evidence that contains unknown dependencies.
- Incorporate evidence that contains long-range effects.
Conditional Markov Random Fields (CRF) replace GHMMs.

- Incorporate evidence that contains unknown dependencies.
- Incorporate evidence that contains long-range effects.
- No need to model the observation data.
Conditional Markov Random Fields (CRF) replace GHMMs.

- Incorporate evidence that contains unknown dependencies.
- Incorporate evidence that contains long-range effects.
- No need to model the observation data.
- Might make more accurate predictions.
Conditional Markov Random Fields (CRF) replace GHMMs.

- Incorporate evidence that contains unknown dependencies.
- Incorporate evidence that contains long-range effects.
- No need to model the observation data.
- Might make more accurate predictions.
- Discriminative modelling: $Pr(Y|X)$. 
Conditional Markov Random Fields (CRF) replace GHMMs.

- Incorporate evidence that contains unknown dependencies.
- Incorporate evidence that contains long-range effects.
- No need to model the observation data.
- Might make more accurate predictions.
- Discriminative modelling: $Pr(Y|X)$.
- GHMM is generative modelling: $Pr(Y, X)$. 
Conditional Markov Random Fields (CRF) replace GHMMs.

- Incorporate evidence that contains unknown dependencies.
- Incorporate evidence that contains long-range effects.
- No need to model the observation data.
- Might make more accurate predictions.
- Discriminative modelling: $Pr(Y|X)$.
- GHMM is generative modelling: $Pr(Y, X)$.
- Conrad is an implementation of SMCRF with CML and MEA training algorithms.
Conditional Markov Random Fields (CRF) replace GHMMs.

- Incorporate evidence that contains unknown dependencies.
- Incorporate evidence that contains long-range effects.
- No need to model the observation data.
- Might make more accurate predictions.
- Discriminative modelling: $Pr(Y|X)$.
- GHMM is generative modelling: $Pr(Y, X)$.
- Conrad is an implementation of SMCRF with CML and MEA training algorithms.
- See CRAIG too!
... and it works!

- On *Cryptococcus neoformans* outperforms TWINSCAN, most accurate trained for C. neoformans.

We owe this to

- Discriminative methods vs generative methods.
- Ability to incorporate additional data.
... and it works!

- On *Cryptococcus neoformans* outperforms TWINSCAN, most accurate trained for C. neoformans.
- On *Aspergillus nidulans* outperforms Fgenesh, used in GenBank annotation.

We owe this to

- Discriminative methods vs generative methods.
- Ability to incorporate additional data.
A sequence $X_0, X_1, X_2, \cdots$ of random variables taking value in the set $S$. They satisfy (i) Markovian property, and (ii) time homogeneity.

$$X = (S, P, \pi).$$

**Example**

Gambler’s fortune: $X_i$ denotes gambler’s fortune at time $i$. Each time he wins $1$ if coin lands $H$ and loses $1$ if it lands $T$. $S = \{0, 1, \ldots, s\}$, cannot play if fortune is $0$ or $s$ dollars. Initially $X_0 = 2$, assume $s > 2$. 
Yes, but what does this have to do with DNA?

Example

A DNA sequence can be modelled by a Markov chain (first-order dependency)

TTTACACATAGATAGAT

$X = (S, P, \pi)$ where

- $S = \{A, C, G, T\}$
- $P = \begin{pmatrix}
    p_{AA} & p_{AC} & p_{AG} & p_{AT} \\
    \ldots & \ldots & \ldots & \ldots \\
    \ldots & \ldots & \ldots & \ldots \\
    p_{TA} & p_{TC} & p_{TG} & p_{TT}
\end{pmatrix}$
- $\pi = (1/4, 1/4, 1/4, 1/4)^t$, uniform distribution.

Higher-order Markov modelling is possible too!
Hidden Markov Models

Definition
An HMM is a finite state (i.e. $|S| < \infty$) Markov chain with outputs.

Example

- $S = \{s_1, s_2\}$
- $P = \begin{pmatrix} 0.9 & 0.1 \\ 0.8 & 0.2 \end{pmatrix}$
- $\pi = (1/2, 1/2)^t$
- $A = \{1, 2\}$
- $S_1$ has uniform distribution on $A$ and $S_2$ emits 1 with prob. $1/4$ and 2 with prob. $3/4$. 
Example continued

Suppose we observe $O = 222$.

- **Question:** What sequence of states has the highest probability given $O$?
Suppose we observe $\mathcal{O} = 222$.

▶ **Question:** What sequence of states has the highest probability given $\mathcal{O}$?

▶ **Answer:** $S_2 S_1 S_1$
So an HMM is a tuple \((S, A, P, B, \pi)\) where

- \(S = \{s_1, s_2, \ldots, s_N\}\) is a finite set of states.
- \(A = \{a_1, a_2, \ldots, a_M\}\) is the output alphabet.
- \(P\) is the transition probability matrix
- \(B\) is the emission probability matrix: \(b_{ij}\) is the prob. that state \(s_i\) emits \(a_j\).
- \(\pi\) initial distribution.

\(\lambda = (P, B, \pi)\) is called the \textit{parameters}, and \((S, A)\) is the topology.
Three questions

Given $\mathcal{O} = O_1 O_2 \cdots O_T$,

- Given $\lambda$, find $Pr(\mathcal{O}|\lambda)$.  

Assuming fixed structure, find $\lambda$ that maximizes $Pr(\mathcal{O}|\lambda)$. 

Esfandiar Haghverdi

An Introduction to Conrad
Three questions

Given $\mathcal{O} = O_1 O_2 \cdots O_T$,

- Given $\lambda$, find $Pr(\mathcal{O}|\lambda)$.
- Find $\arg\max_Q Pr(Q|\mathcal{O})$. 

Esfandiar Haghverdi
An Introduction to Conrad
Three questions

Given $\mathcal{O} = \mathcal{O}_1 \mathcal{O}_2 \cdots \mathcal{O}_T$,

- Given $\lambda$, find $Pr(\mathcal{O} | \lambda)$.
- Find $\text{argmax}_Q Pr(Q | \mathcal{O})$.
- Assuming fixed structure, find $\lambda$ that maximizes $Pr(\mathcal{O} | \lambda)$. 
Three algorithms

▶ Forward algorithm.
Three algorithms

- Forward algorithm.
- Viterbi algorithm, first finds $\max_Q Pr(Q|O)$ and the backtracks to find one such $Q$. 
Three algorithms

- Forward algorithm.
- Viterbi algorithm, first finds $\max_Q \Pr(Q|O)$ and the backtracks to find one such $Q$.
- Baum-Welch method.
Applications

- Modeling protein families, see Krogh et al., profile HMMs.
Applications

- Modeling protein families, see Krogh et al., profile HMMs.
- Multiple sequence alignment.
Applications

- Modeling protein families, see Krogh et al., profile HMMs.
- Multiple sequence alignment.
- Pfam from Sanger center.
Applications

- Modeling protein families, see Krogh et al., profile HMMs.
- Multiple sequence alignment.
- Pfam from Sanger center.
- Gene finding, GENSCAN, see Burge et al.
semi-HMM or more logically Hidden semi-Markov Models (GHMM): Self Transition prob. are zero, emits sequences on A, whose length can follow any distribution. AUGUSTUS, EHMM, GENSCAN, Fgenesh, GeneID.
More and beyond ...

- semi-HMM or more logically Hidden semi-Markov Models (GHMM): Self Transition prob. are zero, emits sequences on $A$, whose length can follow any distribution. AUGUSTUS, EHMM, GENSCAN, Fgenesh, GeneID.

- Phylo-HMM: they improve phylogenetic models allowing for variation among sites in the rate of substitution, used in secondary structure prediction, detection of recombination events, multi-species version of ab initio gene prediction problem.

For more, see Siepel and Haussler.
More and beyond ...

- semi-HMM or more logically Hidden semi-Markov Models (GHMM): Self Transition prob. are zero, emits sequences on A, whose length can follow any distribution. AUGUSTUS, EHMM, GENSCAN, Fgenesh, GeneID.

- Phylo-HMM: they improve phylogenetic models allowing for variation among sites in the rate of substitution, used in secondary structure prediction, detection of recombination events, multi-species version of ab initio gene prediction problem.

  **Same as HMM, but emits an observable drawn from distributions defined by phylogenetic models. A machine that probabilistically generates a multiple alignment column by column, each column defined by a phylogenetic model.**
More and beyond ...

- semi-HMM or more logically Hidden semi-Markov Models (GHMM): Self Transition prob. are zero, emits sequences on A, whose length can follow any distribution. AUGUSTUS, EHMM, GENSCAN, Fgenesh, GeneID.

- Phylo-HMM: they improve phylogenetic models allowing for variation among sites in the rate of substitution, used in secondary structure prediction, detection of recombination events, multi-species version of ab initio gene prediction problem.

  *Same as HMM, but emits an observable drawn from distributions defined by phylogenetic models. A machine that probabilistically generates a multiple alignment column by column, each column defined by a phylogenetic model.*

- For more, see Siepel and Haussler.
Markov Random Fields


➤ Goes back to statistical physics (Ising model, 1925 ferromagnetic material), generalization of Markov processes time index replaced by space index.

\[ Pr(w) = \frac{1}{Z} \exp\left(-\frac{1}{kT}U(w)\right), \text{ Gibbs measure} \]

where \( Z = \sum_w \exp\left(-\frac{1}{kT}U(w)\right) \): partition function, \( U(w) \): energy associated with configuration \( w \).
Markov Random Fields


▶ Goes back to statistical physics (Ising model, 1925 ferromagnetic material), generalization of Markov processes time index replaced by space index.

\[ Pr(w) = \frac{1}{Z} \exp \left( -\frac{1}{kT} U(w) \right), \text{ Gibbs measure} \]

where \( Z = \sum_w \exp \left( -\frac{1}{kT} U(w) \right) \): partition function, \( U(w) \): energy associated with configuration \( w \).

▶ The underlying structure is a linear chain, but can be a lattice or any graph!
Markov Random Fields


- Goes back to statistical physics (Ising model, 1925 ferromagnetic material), generalization of Markov processes time index replaced by space index.

\[ Pr(w) = \frac{1}{Z} \exp \frac{-1}{kT} U(w), \text{ Gibbs measure} \]

where \( Z = \sum_w \exp \frac{-1}{kT} U(w) \): partition function, \( U(w) \): energy associated with configuration \( w \).

- The underlying structure is a linear chain, but can be a lattice or any graph!

- Gibbs measure maximizes the entropy and has Markovian property ...
Let $G = (\mathcal{V}, E)$ be an (undirected) graph. Associate a random variable $X_v$ with each node $v \in \mathcal{V}$, all $X_v$ take values in a set $S$. $N_v$: the set of neighbors of $v$.

Definition
$
\{X_v\}_{v \in \mathcal{V}}$ defines a Markov random field if

$$
Pr(x_v|x_{\mathcal{V} - v}) = Pr(x_v|x_{N_v}).
$$

Example
Any two-sided Markov chain is an MRF. $G$ is just a linear chain.
A potential $V$ is a way of assigning numbers to $V_A(\vec{x})$ where $A \subseteq \mathcal{V}$.

Define energy $U(\vec{x}) = -\sum_A V_A(\vec{x})$ for all $A \subseteq \mathcal{V}$.

Gibbs measure induced by $U$: $Pr(\vec{x}) = \frac{1}{Z} \exp(-U(\vec{x}))$ where $Z = \sum_{w} \exp(-U(w))$ is the partition function.

Nearest neighbor Gibbs potential: $V_A(\vec{x}) = 0$, if $A$ is not a clique in $G$.

$$Pr(\vec{x}) = \frac{1}{Z} \exp\left(\sum_{C} V_C(\vec{x})\right),$$

where $C$ is a clique in $G$. 
Origins in information extraction and computational linguistics, sequential data labelling. See Lafferty et al., Sutton and McCallum. Given a sentence in English, tag it with part-of-speech taggings. $X$ denotes the random variable over English sentences (observation), $\vec{Y}$ (labels, hidden states) family of random variables over taggings, where each $Y_i$ ranges over the finite set of possible tags. For example, $Y_i$ might be “proper noun”.

Definition
A CRF is a random field globally conditioned on a random variable $X$ (observation).

I.e., Given a graph $G = (V, E)$, $\vec{Y} = \{Y_v\}_{v \in V}$. $(X, \vec{Y})$ is a CRF if $(\vec{Y}|X)$ is a random field.

Note that we do not need to model $Pr(X)$, nor $Pr(X, \vec{Y})$, just $Pr(\vec{Y}|X)$.

A linear chain CRF is one where $G$ is a linear (finite) chain.
Main Theorem (Hammersley-Clifford)

\[
Pr(y|x) = \frac{1}{Z_w(x)} \exp \left( \sum_{\text{features } j} w_j F_j(y, x) \right)
\]

- \( F_j \): feature sum
- \( w_j \): weight for feature sum \( F_j \).
Main Theorem (Hammersley-Clifford)

\[
Pr(y|x) = \frac{1}{Z_w(x)} \exp \left( \sum_{\text{features } j} w_j F_j(y, x) \right)
\]

- \( F_j \): feature sum
- \( w_j \): weight for feature sum \( F_j \).
- \( y = y_1 \ldots y_n \), labels like “exon”, “intron”.
Main Theorem (Hammersley-Clifford)

\[ Pr(y|x) = \frac{1}{Z_w(x)} \exp \left( \sum_{\text{features } j} w_j F_j(y, x) \right) \]

- \( F_j \): feature sum
- \( w_j \): weight for feature sum \( F_j \).
- \( y = y_1 \ldots y_n \), labels like “exon”, “intron”.
- \( y = (t_i, u_i, v_i)_{i=1}^p \) of start, stop and label triples (semi-Markov). There are some constraints:
Main Theorem (Hammersley-Clifford)

\[ Pr(y|x) = \frac{1}{Z_w(x)} \exp \left( \sum_{\text{features } j} w_j F_j(y, x) \right) \]

- **F_j**: feature sum
- **w_j**: weight for feature sum \( F_j \).
- \( y = y_1 \ldots y_n \), labels like “exon”, “intron”.
- \( y = (t_i, u_i, v_i)_{i=1}^p \) of start, stop and label triples (semi-Markov). There are some constraints:
  - \( t_1 = 1, u_i \geq t_i, u_{i-1} + 1 = t_i, u_p = n \)
Main Theorem (Hammersley-Clifford)

\[ Pr(y|x) = \frac{1}{Z_w(x)} \exp \left( \sum_{\text{features } j} w_j F_j(y, x) \right) \]

- \( F_j \): feature sum
- \( w_j \): weight for feature sum \( F_j \).
- \( y = y_1 \ldots y_n \), labels like “exon”, “intron”.
- \( y = (t_i, u_i, v_i)^P \) of start, stop and label triples (semi-Markov). There are some constraints:
  - \( t_1 = 1, u_i \geq t_i, u_{i-1} + 1 = t_i, u_P = n \)
  - \( v_{i-1} \neq v_i \) and \( y_{t_i} = y_{t_i+1} = \cdots = y_{u_i} = v_i \)
Main Theorem (Hammersley-Clifford)

\[
Pr(y|x) = \frac{1}{Z_w(x)} \exp \left( \sum_{\text{features } j} w_j F_j(y, x) \right)
\]

- \(F_j\): feature sum
- \(w_j\): weight for feature sum \(F_j\).
- \(y = y_1 \ldots y_n\), labels like “exon”, “intron”.
- \(y = (t_i, u_i, v_i)_{i=1}^p\) of start, stop and label triples (semi-Markov). There are some constraints:
  - \(t_1 = 1, u_i \geq t_i, u_{i-1} + 1 = t_i, u_p = n\)
  - \(v_{i-1} \neq v_i\) and \(y_{t_i} = y_{t_i+1} = \cdots = y_{u_i} = v_i\)
  - restrictions on allowed \(v_{i-1} \rightarrow v_i\) for plausible gene structure, for example “intergenic” to “exon” only at ATG start codon, etc.
Main Theorem (Hammersley-Clifford)

\[ Pr(y|x) = \frac{1}{Z_w(x)} \exp \left( \sum_{features \ j} w_j F_j(y, x) \right) \]

- \( F_j \): feature sum
- \( w_j \): weight for feature sum \( F_j \).
- \( y = y_1 \ldots y_n \), labels like “exon”, “intron”.
- \( y = (t_i, u_i, v_i)^p_{i=1} \) of start, stop and label triples (semi-Markov). There are some constraints:
  - \( t_1 = 1, u_i \geq t_i, u_{i-1} + 1 = t_i, u_p = n \)
  - \( v_{i-1} \neq v_i \) and \( y_{t_i} = y_{t_{i+1}} = \cdots = y_{u_i} = v_i \)
  - restrictions on allowed \( v_{i-1} \rightarrow v_i \) for plausible gene structure, for example “intergenic” to “exon” only at ATG start codon, etc.
- \( F_j(y, x) = \sum_{i=1}^p f_j(v_{i-1}, t_i, u_i, v_i, x) \), sum of localized feature functions.
Using SMCRF in gene prediction

Key issues:

▶ Design of feature functions $f_j$ which use $X$ to capture properties of $X$ relevant to the classification. Observation data need not have a probabilistic model need not be independent. Conrad uses phylo-GHMM for a core feature set, and defines some discriminative features to capture information from phylogenetic footprinting, insertion/deletion events in multiple alignments and EST alignments.

▶ Selection of weights $w_j$: Conrad uses two training algorithms CML and MEA.
Conditional probabilities of GHMM are equivalent to an SMCRF using a single feature with weight=1.0.
Conditional probabilities of GHMM are equivalent to an SMCRF using a single feature with weight=1.0.

Or, as done in the paper using many features all with weight=1.0, here they use 22 features.
Conditional probabilities of GHMM are equivalent to an SMCRF using a single feature with weight=1.0.
Or, as done in the paper using many features all with weight=1.0, here they use 22 features.
ConradG-1 ~ GHMM
Conditional probabilities of GHMM are equivalent to an SMCRF using a single feature with weight=1.0.

Or, as done in the paper using many features all with weight=1.0, here they use 22 features.

ConradG-1 $\sim$ GHMM

ConradG-n, $n \geq 2$ $\sim$ phylo-GHMM using $n$ species.
Training: Conditional Maximum Likelihood

\[ w_{CML} = \arg \max_w \log(Pr_w(Y^0|X^0)) \]

where \((Y^0, X^0)\) is the training data.

- Uses gradient-based function optimizer.
Training: Conditional Maximum Likelihood

\[ w_{CML} = \arg\max_w (\log(Pr_w(Y^0|X^0))) \]

where \((Y^0, X^0)\) is the training data.

- Uses gradient-based function optimizer.
- For SMCRF involves dynamic programming.
Training: Conditional Maximum Likelihood

\[ w_{CML} = \arg\max_w \left( \log \left( Pr_w(Y^0|X^0) \right) \right) \]

where \((Y^0, X^0)\) is the training data.

- Uses gradient-based function optimizer.
- For SMCRF involves dynamic programming.
- Linear computation time in the length of training data and the longest allowed interval.
Training: Conditional Maximum Likelihood

\[ w_{\text{CML}} = \arg \max_w \log \left( \Pr_w (Y^0 | X^0) \right) \]

where \((Y^0, X^0)\) is the training data.

- Uses gradient-based function optimizer.
- For SMCRF involves dynamic programming.
- Linear computation time in the length of training data and the longest allowed interval.
- ConradC-n \((n \geq 1)\) features same as in ConradG-n but uses CML training.
Training: Maximum Expected Accuracy

\[ w_{MEA} = \arg\max_w (A_{MEA}(w)) \]

where

\[ A_{MEA}(w) = E_w(S(Y, Y^0, X^0)) = \sum_y \Pr_w(y|X^0)S(y, Y^0, X^0), \text{ and} \]

\[ S(y, Y^0, X^0) = \sum_{n}^{\text{number of nucs called correctly}} + 200(\text{number of splices called correctly}). \]
Training: Maximum Expected Accuracy

\[ w_{MEA} = \arg\max_w (A_{MEA}(w)) \]

where

\[ A_{MEA}(w) = E_{w}(S(Y, Y^0, X^0)) = \sum_y Pr_w(y|X^0) S(y, Y^0, X^0), \] and

\[ S(y, Y^0, X^0) = \sum_{i=1}^n s(y_{i-1}, y_i, y_{i-1}^0, y_i^0, X^0, i), \] measures the similarity between \( y = y_0 y_1 \cdots y_n \) and \( Y^0 \), given \( X^0 \).
Training: Maximum Expected Accuracy

\[ w_{MEA} = \arg\max_w (A_{MEA}(w)) \]

where

- \( A_{MEA}(w) = E_w(S(Y, Y^0, X^0)) = \sum_y P_{rw}(y|X^0)S(y, Y^0, X^0) \), and
- \( S(y, Y^0, X^0) = \sum_{i=1}^n s(y_{i-1}, y_i, Y^0, y_i, X^0, i) \), measures the similarity between \( y = y_0y_1\cdots y_n \) and \( Y^0 \), given \( X^0 \).
- For example, \( S_{SPlice} = \) number of nucs called correctly + 200(number of splices called correctly).
Training: Maximum Expected Accuracy

\[ w_{MEA} = \arg\max_w(A_{MEA}(w)) \]

where

- \( A_{MEA}(w) = E_w(S(Y, Y^0, X^0)) = \sum_y Pr_w(y|X^0)S(y, Y^0, X^0) \), and

- \( S(y, Y^0, X^0) = \sum_{i=1}^n s(y_{i-1}, y_i, y_{i-1}^0, y_i^0, X^0, i) \), measures the similarity between \( y = y_0y_1\cdots y_n \) and \( Y^0 \), given \( X^0 \).

- For example, \( S_{SPLICE} = \) number of nucs called correctly + 200(number of splices called correctly).

- Problem is we may not be able to find the global maximum, so the initial weights are set using CML training.
SMCRF can incorporate evidence that contain long-range effects, unknown dependencies, or is difficult to model.

- six gap features: capture information from the pattern of gaps in a multiple alignment.
SMCRF can incorporate evidence that contain long-range effects, unknown dependencies, or is difficult to model.

- six gap features: capture information from the pattern of gaps in a multiple alignment.
- For example

\[
f_{\text{GAP,EXON12}}(v_{i-1}, v_i, t_i, u_i, X) = \sum_{k=t_i}^{u_i} c_k(v_i)
\]

where \( c_k(v_i) = 1 \) if \( v_i = \text{exon} \) and the multiple alignment \( X \) has a gap of length 1 or 2 (mod 3) with a boundary at position \( k \), and 0 otherwise.
 Discriminative features continued

- three footprint features per species indicate the positions at which each species is aligned.

\[ \text{FOOT}(\text{v}_{i-1}, \text{v}_i, t_i, u_i, X) = u_i \sum_{k=1}^{t_i} c_k(v_i) \]

where \( c_k(v_i) = 1 \) if \( v_i = \text{exon} \) and species \( S \) is aligned at position \( k \), and 0 otherwise.
Discriminative features continued

- three footprint features per species indicate the positions at which each species is aligned.
- For example,

\[
f_{\text{FOOT,EXON}}(v_{i-1}, v_i, t_i, u_i, X) = \sum_{k=t_i}^{u_i} c_k(v_i)
\]

where \( c_k(v_i) = 1 \) if \( v_i = \text{exon} \) and species \( S \) is aligned at position \( k \), and 0 otherwise.
Nine EST features relate the alignment of ESTs to the states and transitions of the hidden sequence.
Nine EST features relate the alignment of ESTs to the states and transitions of the hidden sequence.

For example,

$$f_{EST, EXON, CONSISTENT}(v_{i-1}, v_i, t_i, u_i, X) = \sum_{k=t_i}^{u_i} c_k(v_i, e_k)$$

where $c_k(v_i, e_k) = 1$ if $v_i = \text{exon}$ and $e_k = \text{exon}$, and 0 otherwise. $e_k$ indicates the EST evidence at position $k$. 
Conrad training

- Given: Model parameter file (a list of features and a training method) + a set of training data
Conrad training

- Given: Model parameter file (a list of features and a training method) + a set of training data
- Conrad: uses training data to learn numerical parameters for the features: intron length distribution, position weight matrix for the splice sites, rates of nucleotide sequence evolution. Then,
Conrad training

- Given: Model parameter file (a list of features and a training method) + a set of training data
- Conrad: uses training data to learn numerical parameters for the features: intron length distribution, position weight matrix for the splice sites, rates of nucleotide sequence evolution. Then,
- trains the feature weights (if CML or MEA is specified) or sets all weights to 1 (produces GHMM)
Experimental design

- C. neoformans, 19 Mb genome, ~ 7000 genes, on average 6 exons per gene.
- EST feature as a positive control.
- A. nidulans, 30 Mb genome, more than 10,000 genes.
- Conrad vs Twinscan and GenelD, most accurate gene predictors trained for C. neoformans. Chromosome 9 and compared against all EST data.
- Conrad vs Fgenesh on A. nidulans, whole genome, compared against all EST data.
Results: training approaches

**Figure 1.** Performance of Conrad models in the C. neoformans cross-validation tests. The graph on the left shows average gene sensitivity (percentage of reference genes completely correct in the testing set) across 10 replicates based on model and training set size. Solid lines are performance on the test set; and dotted lines are performance on the training set (not all training sets are shown). The bars represent standard deviation across the replicates. The table on the right shows the full set of testing accuracy statistics for the models on the 600-gene training sets.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ConradSFGE-5</td>
<td>MEA-splice</td>
<td>5</td>
<td>Gap/Foot</td>
<td>98.8</td>
<td>99.4</td>
<td>95.0</td>
<td>96.7</td>
<td>86.0</td>
<td>86.0</td>
</tr>
<tr>
<td>ConradSFG-5</td>
<td>MEA-splice</td>
<td>5</td>
<td>Gap/Foot</td>
<td>98.1</td>
<td>96.4</td>
<td>89.4</td>
<td>92.4</td>
<td>70.1</td>
<td>70.2</td>
</tr>
<tr>
<td>ConradS-5</td>
<td>MEA-splice</td>
<td>5</td>
<td>—</td>
<td>98.0</td>
<td>98.1</td>
<td>87.3</td>
<td>91.1</td>
<td>64.6</td>
<td>64.9</td>
</tr>
<tr>
<td>ConradSFG-2</td>
<td>MEA-splice</td>
<td>2</td>
<td>Gap/Foot</td>
<td>98.3</td>
<td>96.3</td>
<td>88.1</td>
<td>91.2</td>
<td>66.5</td>
<td>66.1</td>
</tr>
<tr>
<td>ConradC-2</td>
<td>CML</td>
<td>2</td>
<td>—</td>
<td>97.9</td>
<td>98.2</td>
<td>85.6</td>
<td>90.2</td>
<td>61.3</td>
<td>61.8</td>
</tr>
<tr>
<td>ConradN-2</td>
<td>MEA-Nucleotide</td>
<td>2</td>
<td>—</td>
<td>98.0</td>
<td>96.1</td>
<td>86.0</td>
<td>90.0</td>
<td>61.1</td>
<td>61.0</td>
</tr>
<tr>
<td>ConradS-2</td>
<td>MEA-splice</td>
<td>2</td>
<td>—</td>
<td>97.8</td>
<td>98.1</td>
<td>85.8</td>
<td>89.9</td>
<td>60.8</td>
<td>60.8</td>
</tr>
<tr>
<td>ConradG-2</td>
<td>GHMM</td>
<td>2</td>
<td>—</td>
<td>94.7</td>
<td>98.4</td>
<td>82.3</td>
<td>87.0</td>
<td>52.7</td>
<td>54.1</td>
</tr>
<tr>
<td>ConradS-1</td>
<td>MEA-splice</td>
<td>1</td>
<td>—</td>
<td>97.7</td>
<td>97.3</td>
<td>81.5</td>
<td>86.6</td>
<td>53.3</td>
<td>53.3</td>
</tr>
</tbody>
</table>

* Models in table are divided into three sets according to the input data they use: the top group uses all available data; the second uses only a two-species alignment; and the third uses only the reference genome.
Three groups of data.

Compare middle group for gene and exon sensitivity and specificity. Generative vs discriminative training.

Discriminative training methods show very similar performance.

Max accuracy reached at $\sim 600$-$800$ genes (training set)

Dotted lines: ConradSFG-5 and ConradC-2 on training set.

discriminative training is more expensive (MEA 8-12 CPU-h/Mb, CML 6-8 CPU-h/Mb) than generative training (GHMM 39 CPU-s/Mb).

But, inference algorithms take the same time (8.5 min/Mb)
Results: Improved accuracy due to features

Going back to figure!

- amazing change form ConradSFG-5 to ConradSFGGE-5 (one must avoid overtraining)
Table 2. Performance of Conrad and other gene callers on *C. neoformans* chromosome 9

<table>
<thead>
<tr>
<th>Model</th>
<th>Training method</th>
<th>No. of species</th>
<th>Additional features</th>
<th>No. of genes</th>
<th>No. of exons</th>
<th>Avg. gene length (bases)</th>
<th>Genes with EST overlap</th>
<th>Consistency (%)</th>
<th>Missed ESTs</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenelD</td>
<td>Generative</td>
<td>1</td>
<td>N/A</td>
<td>472</td>
<td>2719</td>
<td>1490</td>
<td>260 (55%)</td>
<td>40.8</td>
<td>86</td>
</tr>
<tr>
<td>ConradS-1</td>
<td>MEA-splice</td>
<td>1</td>
<td>—</td>
<td>489</td>
<td>2876</td>
<td>1547</td>
<td>264 (54%)</td>
<td>63.3</td>
<td>85</td>
</tr>
<tr>
<td>Twinscan</td>
<td>Generative</td>
<td>2</td>
<td>N/A</td>
<td>478</td>
<td>2760</td>
<td>1477</td>
<td>267 (56%)</td>
<td>65.5</td>
<td>95</td>
</tr>
<tr>
<td>ConradG-2</td>
<td>Generative</td>
<td>2</td>
<td>N/A</td>
<td>486</td>
<td>2826</td>
<td>1437</td>
<td>271 (56%)</td>
<td>65.7</td>
<td>96</td>
</tr>
<tr>
<td>ConradC-2</td>
<td>Cond. ML</td>
<td>2</td>
<td>—</td>
<td>469</td>
<td>2823</td>
<td>1595</td>
<td>263 (56%)</td>
<td>66.9</td>
<td>87</td>
</tr>
<tr>
<td>ConradN-2</td>
<td>MEA-nuc.</td>
<td>2</td>
<td>—</td>
<td>487</td>
<td>2823</td>
<td>1522</td>
<td>269 (55%)</td>
<td>69.5</td>
<td>88</td>
</tr>
<tr>
<td>ConradS-2</td>
<td>MEA-splice</td>
<td>2</td>
<td>—</td>
<td>477</td>
<td>2823</td>
<td>1544</td>
<td>266 (56%)</td>
<td>71.1</td>
<td>87</td>
</tr>
<tr>
<td>ConradSFG-2</td>
<td>MEA-splice</td>
<td>2</td>
<td>Gap/Foot</td>
<td>477</td>
<td>2855</td>
<td>1564</td>
<td>264 (55%)</td>
<td>78.4</td>
<td>87</td>
</tr>
<tr>
<td>ConradS-5</td>
<td>MEA-splice</td>
<td>5</td>
<td>—</td>
<td>469</td>
<td>2837</td>
<td>1595</td>
<td>262 (56%)</td>
<td>76.0</td>
<td>88</td>
</tr>
<tr>
<td>ConradSFG-5</td>
<td>MEA-splice</td>
<td>5</td>
<td>Gap/Foot</td>
<td>465</td>
<td>2857</td>
<td>1601</td>
<td>261 (56%)</td>
<td>82.4</td>
<td>88</td>
</tr>
<tr>
<td>ConradSFGE-5</td>
<td>MEA-splice</td>
<td>5</td>
<td>Gap/Foot/EST</td>
<td>510</td>
<td>2881</td>
<td>1421</td>
<td>275 (54%)</td>
<td>94.5</td>
<td>104</td>
</tr>
</tbody>
</table>

The models are grouped by the evidence they used as input: single gene sequence, pairwise alignment, and then all available data. Predictions were compared to available EST (expressed sequence tag) evidence using a custom set of metrics designed to handle partial information from ESTs. Shown are the total number of genes and exons predicted by each model and the number of those predictions that overlapped EST evidence. Of those predictions overlapping EST evidence, the percent where the EST and gene predictions agree is shown. Also included is the total number of EST clusters that did not overlap any prediction, indicating probable missed genes.
- ConradSFG-2 vs Twinscan
- ConradG-2 vs Twinscan
- ConradS-2 vs ConradG-2
- ConradSFG-2 vs ConradS-2
- ConradSFG-5 vs ConradSFG-2.
- ConradSFGE-5 vs ConradSFG-5.
Aspergillus clade has nine fully sequenced genomes.

**Figure 2.** Accuracy results for the ConradSGF-n model on *A. nidulans* using several different combinations of informant species. Branch lengths are in substitutions per site based on a set of highly conserved housekeeping genes.
300-gene test set and 274-gene training set, all with ConradSGF-n

Look at single species and two species, distance issue.
Discussion

- An implementation of SMCRF.
- State of the art gene annotation tool in fungi.
- Discriminative vs generative training.
- Easy incorporation of additional information, e.g., gap and footprint, informant species, EST.
- Has been incorporated into the Broad institute pipeline for eukaryotic genomes.
- highly customizable.
Future work

- Improve feature sets, here it was seeded by generative features (phylo-GHMM).
- Improve accuracy and extend to mammalian genomes, arbitrarily long introns and intergenic regions, include UTRs, parallelization.
- Incorporate long-range biological interactions, e.g. upstream CpG islands, exonic splicing enhancers, chromatin methylation patterns, etc.
- Negative examples for training sets.
- Other and perhaps specialized training algorithms to improve accuracy.