Predicting V(D)J Recombination Using Conditional Random Fields

Raunaq Malhotra, Shruthi Prabhakara, and Raj Acharya

Department of Computer Science Engineering, Pennsylvania State University, University Park, PA, 16801, USA
{rom5161,sap263,acharya}@cse.psu.edu

Abstract. V(D)J gene segments undergo combinatorial recombination in the T-cells and B-cells to provide humans and other vertebrates with a large number of antibodies required for immunity. Each such recombination further undergoes mutations in their DNA sequences so that they can recognize diverse antigens. Predicting the combination of gene segments which formed a particular antibody is an essential task for studying disease propagation and analysis. We propose a model based on conditional random fields (CRFs) for predicting the boundary positions between V-D-J gene segments. We train the CRFs by generating synthetic gene recombinations using all of the alleles of the V, D and J gene segments. The alleles corresponding to a read can be determined by mapping the segmented reads to the DNA sequences of the gene segments using softwares like BLAST and usearch. We test our method on simulated dataset as well as real data of Stanford_S22 individual.

Keywords: Conditional Random Fields, VDJ recombination, Mapping of DNA sequences.

1 Introduction

The immune system of an organism provides protection against a wide range of antigens with the help of a large number of antibodies. These antibodies are encoded from genes within the B-cells, and bind to different antigens in order to protect organisms from diseases. The large number of genes that encode these antibodies are primarily produced by combinatorial recombination of gene segments within the B-cells. Identifying the gene segments which encode for a particular antibody is important for understanding the immune response to different types of antigens, and in the study of infections.

In B-cells, three types of gene segments or germline components, namely variable (V), diversity (D) and joining (J), combine together to form the variable region of the immunoglobulin gene [10]. This combination of gene segments takes place in a combinatorial fashion, in which one of the many alleles of D gene segment combines with an allele of J gene segment. This complex then combines with one of the alleles of V gene segment to form a rearranged gene, which has deleted segments between the joined regions. This process of combinatorial

T. Shibuya et al. (Eds.): PRIB 2012, LNBI 7632, pp. 210–221, 2012.

recombination is known as the VDJ recombination. These antibodies can undergo somatic mutations in their DNA sequences by a process known as somatic hyper-mutation [18].

Each antibody molecule consists of light and heavy chain protein molecules [17]. The heavy chain molecule is made up of a VDJ recombination while the light chain consists of recombinations of V and J gene segments only. In humans, there are 281 V, 84 D and 12 J heavy chain alleles [20], which can produce 283,248 possible heavy chain molecules. The number of known functional heavy chain alleles, however, are lesser (50 for V, 27 for D, and 6 for J giving 8100 possible heavy chain molecules [13]). Two types of light chains are also known, the κ [15] and λ [6]. Thus, only considering the combinatorial rearrangements, there can be millions of possible antibodies.

A host of methods have been proposed that align the sequences to the germline gene segments in order to determine the V(D)J configuration. IMGT/V-QUEST maps the DNA sequences of the antibody to an immunoglobulin and T-cell database to identify the V, D and J alleles [8]. JOINSOLVER, on the other hand, determines the gene segments by identifying the conserved motifs in the target gene [20]. SoDA implements a 3-D lattice alignment based on dynamic programming to traverse through all possible states of VDJ gene segments to determine the single highest scoring alignment [21]. The above methods do not provide a meaningful way of evaluating different rearrangements. Moreover, the large number of possible configurations makes sequence alignment equally time consuming and computationally intensive.

iHMMune-align is a probabilistic model that uses Hidden Markov Models (HMMs) for modeling the genes of an antibody to determine their constituent gene segments [7]. The software creates an HMM model for each of the V gene segment alleles connected to all the possible D and J gene segments. It also models the N-nucleotide additions and exonuclease action around the V-to-D or D-to-J gene segment boundaries. Soda2 is another HMMs based statistical model [17]. Although HMMs have been used efficiently for sequential data tasks, a HMM only models the dependencies between a base and its preceding context. It assumes the distribution to be independent of bases in subsequent positions, given the preceding context. Also the transition probability between two states in an HMM are independent of the bases observed in the two states. Such assumptions reduces the model complexity and makes the model tractable. However, in a typical gene segment, the distribution of bases is dependent throughout the length of the sequence, rendering such assumptions invalid.

In this paper, we propose a model based on CRFs that takes into account such dependencies without increasing the inference computation drastically. CRFs are a special type of Markov random fields where the unknown output variables are conditioned on the input variables [12]. For gene allele prediction, as each gene is a combinatorial recombination of the V, D, and J gene segments, the task at hand is to predict the boundary between the gene segments that make up an antibody. First, we predict the boundary between V and D, using a consensus of all V and D alleles in the database. Next, we infer the specific configuration of V

and D allele by mapping the segment before the boundary to the known alleles of V and after the boundary to the alleles of D. An identical process is followed for inferring the boundary between D and J gene segments and the corresponding J allele.

The CRFs are trained on a dataset of rearranged VDJ gene segments, where the boundary positions between the gene segments are known. After training, when given a DNA sequence, the CRF predicts a label for each base in the DNA sequence. The label for each base indicates the gene segment (V, D or J) that generated the corresponding base. The alleles constituting the DNA sequence can be determined by mapping the segmented DNA sequence to the database of known alleles.

The paper is organized as follows. Section 2 describes the method based on CRFs for predicting the label sequence corresponding to the input DNA sequence. Section 3 explains the experimental setup and results obtained for simulated dataset. We conclude the paper with a summary and a discussion of future extensions of this work.

2 Methods

We are given a set $\mathbf{X} = \{X_1, X_2, ...X_N\}$ of N reads, each of which is sampled from the rearranged genes. Here each of read is of the form $X_i = \{x_{i1}x_{i2}...x_{in}\}$ where $x_i \in \{A, G, C, T\}$. The read length n may vary from read to read. Our objective is to associate each read X_i with a sequence of labels $Y_i = \{y_{i1}y_{i2}...y_{in}\}$, where y_{ik} denotes the gene segment set from which the base x_{ik} was generated. These sets of gene segments are denoted as $\mathbf{V} = \{V_1, V_2, .., V_K\}$, $\mathbf{D} = \{D_1, D_2, ...D_L\}$, and $\mathbf{J} = \{J_1, J_2, ..., J_M\}$, where (K, L, M) denote the number of alleles for corresponding gene segments. Here, V_i, D_i, J_i represent an allele of the corresponding gene segments.

We address the problem of determining the gene segments constituting a read in two steps. In the first step, we identify the bases x_{ik} and x_{il} at which a transition from V-to-D and D-to-J gene segment occurs. If the boundaries are present within the read, we label each of the bases $(x_{i1}x_{i2}...x_{ik})$ as $\mathbf{V}$, the ones between V-to-D and D-to-J boundaries $(x_{ik+1}x_{ik+2}...x_{il})$ as $\mathbf{D}$, and the rest $(x_{il+1}x_{il+2}...x_{in})$ as $\mathbf{J}$. In the second step, we determine the alleles for each gene segment (V_i, D_j, J_k) by mapping the segmented portions of the read labeled $\mathbf{V}$, $\mathbf{D}$ and $\mathbf{J}$ to the corresponding alleles in the immunoglobulin database.

2.1 Conditional Random Fields for Gene Segment Boundary Detection

For the first part, we propose a model based on conditional random fields (CRFs) for predicting the boundaries between the gene segment set that generates a read. Formally, each read $\mathbf{x} = \{x_1x_2...x_n\} \in \mathbf{X}$ is associated with a sequence of labels $\mathbf{y} = \{y_1y_2...y_n\}$ using CRFs. CRFs were originally proposed as probabilistic models for segmentation and sequential labeling[12]. Such methods have

been applied in natural language processing, bioinformatics, image and video segmentation [16,14,1].

We use the linear-chain model of CRFs, where an input node x_i represents a base at a position i in read $\mathbf{x} \in \mathbf{X}$ and an output node y_i denotes the corresponding gene segment label. The conditional probability of the label sequence $\mathbf{y}$ given the observation $\mathbf{x}$ is proportional to

$$\sum_i \exp\left(\sum_j \lambda_j h_j(\mathbf{y}, \mathbf{x}, i)\right) \tag{1}$$

Here $h_j(\mathbf{y}, \mathbf{x}, i)$ is a feature function defined on a subset of the input and output variables that form a clique on the undirected graph and also on the current position i in the input sequence $\mathbf{x}$. The exponential (log-linear) terms in the probability expression are also known as potential functions. For the linear chain graph, where each output label y_i is connected to the preceding output label y_{i-1}, and the input gene sequence $\mathbf{x}$, the feature function is of the form $h_j(y_i, y_{i-1}; \mathbf{x}, i)$. Another popular choice of feature functions are $h_j(y_i; \mathbf{x}, i)$, where the dependence of the current label on the input sequence is captured. These two feature functions are commonly known as the *transition* and *state* feature functions.

The feature functions can be designed to capture various aspects of the given dataset, such as modeling the dependencies on the entire sequence $\mathbf{x}$, as opposed to just the preceding context. This is one of the properties that makes conditional random variables more powerful than Hidden Markov Models for sequential labeling. Each feature function is weighted by λ_j , which determines its contribution in predicting the label. The normalizing constant $Z(\mathbf{x})$ is defined as the sum over all the output labels of all the log-linear potential functions defined above.

$$Z(\mathbf{x}) = \sum_{\mathbf{y}} \sum_i \exp\left(\sum_j \lambda_j h_j(\mathbf{y}, \mathbf{x}, i)\right) \tag{2}$$

Thus, the probability of a label sequence $\mathbf{y}$ given the input sequence $\mathbf{x}$ is given by

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \sum_i \exp\left(\sum_j \lambda_j h_j(\mathbf{y}, \mathbf{x}, i)\right) \tag{3}$$

where $\mathbf{\Lambda} = \{\lambda_j\}$ are the parameters of the model. Given a training dataset D, containing a set $\mathbf{X}$ of N sequences and their labels $\mathbf{Y}$ in a training set, we define a log-likelihood parameterized by Λ over all the training samples as

$$L(\mathbf{\Lambda}) = \sum_{(\mathbf{x},\mathbf{y}) \in \mathbf{D}} \log P(\mathbf{y}|\mathbf{x}) \tag{4}$$

The parameter values that maximize the above likelihood are chosen as the model parameter values. To determine the maximum, one can use gradient ascent methods such as Margin Infused Relaxed Algorithm (MIRA) [4], Limited memory BFGS [3].

The model parameters $\mathbf{\Lambda}$ that maximize the conditional likelihood are used for predicting the sequence of labels for test read $\mathbf{x}^*$ as follows:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}^*) \tag{5}$$

The predicted sequence of labels $\mathbf{y}^*$ indicates the boundaries of the gene segments present in the test sequence.

Feature Functions. The log-linear nature of the feature functions provide the ability to capture complex dependencies on the input data without exponentially increasing the computational complexity for the inference. For applications in text processing such as named entity recognition (NER), the feature functions can be defined to incorporate the grammar of the language, for instance, the word capitalization. In another example, $h_j(\mathbf{y}, \mathbf{x}, i)$ could be defined to count the number of words starting with a capital letter in a sentence. Incorporating feature functions which capture such information increases the predictive power of the model.

In the current context, there is no prior knowledge about such grammar rules for VDJ recombination. In order to overcome such a challenge, we created a set of features which captures different dependencies in the neighborhood of a given base, and learns their weighting parameters from the training dataset. Ideally, the feature functions relevant for determining a V-to-D or a D-to-J junction should get higher weights as compared to the others. The features used are listed in Table 1.

Table 1. Feature functions used for predicting the V,D,J gene segments

Size of neighborhood	Relation to current base
1-base	x_{i-2}
	x_{i-1}
	x_i
	x_{i+1}
	x_{i+2}
2-base	$x_{i-1}x_i$
	x_ix_{i+1}
3-base	$x_{i-2}x_{i-1}x_i$
	$x_ix_{i+1}x_{i+2}$
4-base	$x_{i-3}x_{i-2}x_{i-1}x_i$
	$x_ix_{i+1}x_{i+2}x_{i+3}$
5-base	$x_{i-4}x_{i-3}x_{i-2}x_{i-1}x_i$
	$x_ix_{i+1}x_{i+2}x_{i+3}x_{i+4}$
	$x_{i-2}x_{i-1}x_ix_{i+1}x_{i+2}$

2.2 Boundary Detection and Determination of Gene Segment Alleles

Once we obtain the sequence of labels $\mathbf{y}$ for a given sequence $\mathbf{x}$ using CRFs, we can determine the boundary between V-to-D and D-to-J gene segments as given in $\mathbf{y}$. A base's predicted label is considered to be spurious, if all the neighboring bases within a distance of 4 have a identical labels that is different from that of the base under consideration. We correct for such spurious predictions in our method using mode filtering.

The alleles of gene segment present in a read are determined by mapping boundary segmented parts of reads to their corresponding gene segment set. For example, if a part of the read that is predicted to be generated from **V** gene segment, we map the read to the alleles in the V-gene segment set to determine the closest matching allele V_i. We use a program *usearch* for mapping the sequence on the allele and assign to it the label of the allele with the highest scoring alignment [5].

3 Experiments and Results

First, we evaluate the performance of CRFs in predicting boundaries between gene segments on simulated datasets. We synthetically generated all the combinatorial recombinations of the alleles of gene segments. The allele sequences for V, D and J gene segments in humans, are known. The combinatorial rearrangements of V, D and J alleles are generated by concatenating an allele of V with an allele of D, followed by an allele of J gene segment. In humans, there are 281 V gene segments, 84 D gene segments and 12 J gene segments, giving rise to a total of 283,248 possible recombinations [20]. The downloaded gene segments are from the Kabat database available on the JOINSOLVER website[20]. The statistics of the V, D, and J gene segments are given in Table 2.

Table 2. Statistics of the alleles present in the Kabat gene sequence database

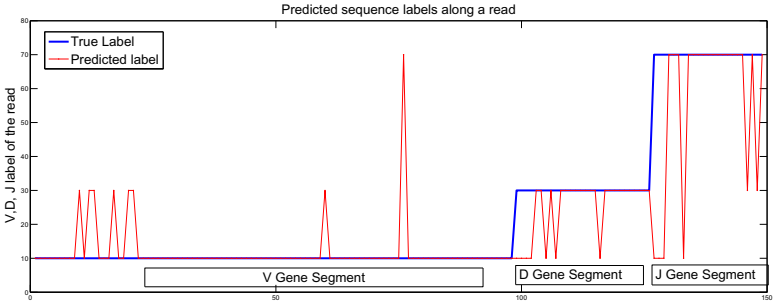
Gene Segment	Total Number	Average Length	Maximum Length	Minimum Length
V gene segments	281	287	305	103
D gene segments	84	25	37	11
J gene segments	12	53	63	48

We randomly choose 60% of these combinatorial rearrangements for training the CRFs, and use the remaining 40% for testing. We repeated the experiment 5 times in which different 60% of the dataset was used for training, and the remaining 40% for testing. For training the CRFs, we used the software package CRF++ [11]. This implementation allows us to select a set of feature functions based on arbitrary combinations of neighboring nucleotides. Table 1 shows the feature functions that were used for training the linear CRF. We use a combination of bi-,tri-,tetra-, and penta-mers to train the CRF. We did not incorporate

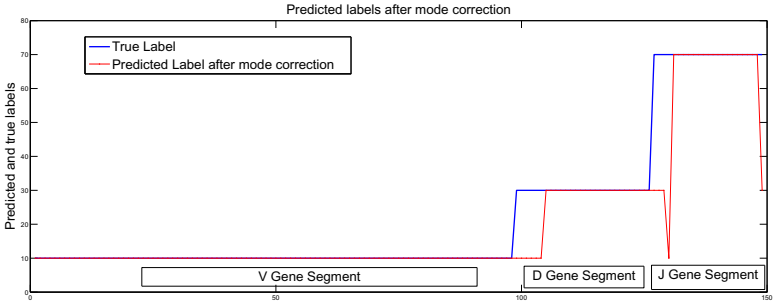
the state transition type feature functions as such prior knowledge is usually not available for a real dataset. For training, the default LBFGS training algorithm in CRF++ was used.

The test data for the boundary prediction by CRFs is generated as follows. We randomly choose 10 combinatorial rearrangements from the 40% of the data not used for training and sample reads using 454 sequencing technology. We used MetaSim to simulate 454 sequencing reads [19] with an average length of 200 bps and standard deviation 20 bps.We simulated reads using the default parameters for 454 sequencing technology provided in MetaSim.

For a given read $\mathbf{x}$, the CRFs model returns a label sequence $\mathbf{y}$ where each label represents the gene segment from which the corresponding base was generated.



(a) Before the mode filtering



(b) After the mode filtering

Fig. 1. Predicted label sequence for one read

After obtaining the label sequence $\mathbf{y}$ for a given read $\mathbf{x}$, we need to predict the boundary positions between the gene segments. We predict a gene segment boundary at a base x_i, if all the bases after x_i are labeled by a different label as compared to the bases before x_i. A base x_i's label prediction y_i is considered to be spurious if it was surrounded by similarly labeled bases, that differ from the

label y_i. For example, we observe that for most of the reads, there is one base labeled as D when all the other surrounding bases in the read are labeled as V. This is depicted in Figure 1, where we represent a read on the x-axis and the true and predicted labels for a read are shown on the y-axis. The V, D, and J labels are assigned levels of 10, 30, and 70 on y-axis for ease of representation.

Predicting a boundary at each position where we observe a change in labeling of the bases in the read, generates a large number of gene segment boundaries, which are not present in the read. We address this problem by first performing a 9-based wide mode filtering on the predicted label sequence. This technique relabels each of base to the mode label in a 9 base window centered on the current base. The window size of 9 was chosen heuristically. A boundary between V-to-D gene segments is called if there is a transition from V-to-D labels in the mode corrected label sequence. If there are multiple such transitions, then we call a boundary at a base having the minimum number of bases labeled as V after the transition. Also, in a given read, as a V-to-J transition is not a valid transition, and we ignore them. We also correct the labeling of all bases between the V-to-D transition and the D-to-J transition as D.

The time complexity for the overall method is same as the time complexity of the CRF method to predict the boundaries for a given set of reads. Once the models for V, D and J gene segments are trained, we can use them for prediction for any number of datasets. The boundary prediction correction, as described above, takes a linear time in terms of the number of reads, thus the time-intensive step being the training time for the CRF method.

Table 3. Precision, Recall, True Negative and Accuracy results for boundary detection of V-to-D and D-to-J gene segments

	V-to-D boundary	D-to-J boundary
Recall	$95.7 \pm 0.8\%$	64.1 ± 7.5
Precision	$64.5 \pm 3.2\%$	93.6 ± 3.9
True Negative	$60.5 \pm 6.7\%$	$98.2 \pm .8$
Accuracy	$75.6 \pm 3.1\%$	88.9 ± 2.1

Table 3 reports the precision and recall rates for predicting a gene boundary averaged over the 5 test datasets. These values are calculated separately for the V-to-D and the D-to-J gene segment boundaries. The precision is defined as the number of reads in which a boundary is correctly detected divided by the total number of reads in which same boundary is detected. The recall rate is defined as the ratio of the number of reads in which the gene boundary is correctly detected to the number of reads which actually have that gene boundary. CRFs are more than 90% precise in detecting the boundary between the D-to-J gene segments and are more than 88% accurate for the same. However, the V-to-D boundary detection is not as precise. This can be attributed to the smaller lengths of the D gene segments, making it difficult to correctly predict a base as D.

For most cases, the gene segment boundary was predicted within 6 bases of the actual boundary. We compute the difference between the base position of a predicted gene segment boundary and the base position of a true gene segment boundary. The percentage of the reads in which the boundary was detected within a k base pairs from the true gene segment boundary is shown in Table 4 for $k = \{2, 3, 4, 5, 6\}$. We report the results separately for V-to-D and D-to-J boundaries. The algorithm predicts the boundary between gene segments within six base pairs with an average accuracy of 80%. One can segment the reads using the predicted boundary positions and map the segmented parts to the corresponding gene segments sets to determine the constituent allele within the read.

Table 4. Boundary prediction results as obtained after performing the mode filtering of the labeled sequences

Base pairs window	V-to-D	D-to-J
2 base pairs	$31.4 \pm 11.2\%$	$32.6 \pm 3.1\%$
3 base pairs	$48.2 \pm 8.7\%$	$46.2 \pm 4.8\%$
4 base pairs	$63.1 \pm 8.1\%$	$61.8 \pm 4.6\%$
5 base pairs	$71.9 \pm 7.5\%$	$69.6 \pm 2.9\%$
6 base pairs	$80.2 \pm 4.6\%$	$73.7 \pm 2.7\%$

Table 5 shows the 5-fold precision and recall values for the gene label prediction on a per base basis. The recall for V (D or J) gene segments is defined as the number of bases across all reads which were correctly identified as V (D, or J) divided by the total number of bases with true labels as V (D or J). The precision value is defined as the number of bases correctly labeled as V (D or J) gene segments divided by the total number of bases labeled as V (D or J) gene segments. We observe the highest precision and recall values for the longer V gene segments and lowest values for shorter D gene segments.

Table 5. Precision and recall values for the predicted gene segments on a per base basis

Gene Segment	Recall	Precision
V gene segments	$91.0 \pm 3.2\%$	$97.5 \pm 0.2\%$
D gene segments	$68.9 \pm 1.2\%$	$35.1 \pm 7.9\%$
J gene segments	$74.2 \pm 0.9\%$	$61.5 \pm 9.2\%$

For testing our models on real transcriptome dataset, we use the CRFs trained on all of the synthetic generated recombinations. As the transcriptome for S22 individual consists of rearranged V,D, J gene segments, and the CRFs are also trained on all the junctions obtained from human V,D and J genes, we believe that the usage of the CRFs trained above are a valid choice for the S22 individual.

The Stanford_S22 dataset consists of 13,153 reads from the rearranged VDJ genes for an individual. These reads were obtained from the DNA sequences derived from peripheral blood mononuclear cells [9]. The genotype of the individual is known through a previous study [2]. Thus, we can use the genotype to evaluate the predictions made by our model. We also compare our error rates with the iHMMune align method [7] mentioned in the benchmarking paper [9].

We used our model trained from all synthetically generated recombinations to predict the labels for each base in the reads of the Stanford_S22 dataset. The gene segment boundaries are determined in a read in a similar fashion to that used in the simulated dataset. Using all the predicted gene positions for a V-to-D (or a D-to-J) transition, we call a V-to-D (or a D-to-J) transition at a position which has the minimum number of V (or D and V) gene labeled bases after the gene position. If a D-to-J transition is absent in a read, we call a D-to-J boundary at a base position which is length of D base pairs after the V-to-D transition. This is easily obtained as the length of the D gene segments are known. We use similar corrections for incorrect prediction of a D-to-J transition before a V-to-D transition. Also, as before, a V-to-J transitions are ignored as they are incorrect.

To evaluate our method, we extract gene segments from each read based on the predicted boundaries. We map the predicted gene segments to the database of V, D and J genes using the software *usearch* [5]. An error in the mapping is counted if the mapped gene is not present in the genotype of the individual (given in the dataset). We compared these error results with that obtained for iHMMune align [7]. Table 6 summarizes our results. The error percentages reported for our method are comparable and even better than that for iHMMune-align. This can be explained on the basis that iHMMune align assumes an inherent Markov chain property where the prediction for a base is dependent on the previous bases only. In contrast a CRF uses potential functions dependent on all types of neighborhood relations between the bases. Also as all the genes of one type are modeled together, the general relationship between the genes of a type is captured in the CRF model. This helps in accurately predicting the boundaries between the gene segments. The relevant gene segments for a gene can be determined based on well established sequence searching algorithms (such as BLAST, usearch) once the boundaries are determined.

Table 6. Comparison of our method (CRF-based) to iHMMune Align. The numbers in the parenthesis are the number of errors for each gene type. The error was called for both using a similar technique.

Gene ID	Error % iHMMune Align	Error % CRF-based
V genes	(707) 5.3%	(136) 1.0%
D genes	(1008) 7.6%	(68) 0.5%
J genes	(10) 0.08%	(18) 0.13%

4 Conclusion and Future Work

We have applied the CRFs for identifying the junctions in VDJ recombination. The approach is very similar to Named Entity Recognition in the text domain. In the text domain, each word is labeled as a named entity or not, in a similar fashion, we label parts of the DNA sequences as belonging to the V, D, or J gene segments. The boundary predictions are within 6 base pairs difference of the actual transition in the simulated data. This is the approximately the number of bases that are deleted and inserted (N-nucleotide additions) when the recombination process happens. Thus our method is predicting the gene boundaries within the accepted accuracy. Our method also works well on the Stanford_S22 dataset, where the boundary predictions made lead to most of the gene segments mapping within the genotype of the individual. It is comparable and in some respects better than iHMMune align for predicting the gene segment boundaries. That being said, our method is a work in progress. We have not considered hyper-mutations of the VDJ recombinations, which often change the DNA sequences of these gene segments. These hyper-mutations introduce an additional challenge in predicting the boundaries between the gene segments. Nevertheless, boundary detection between the gene segments when combined with mapping of the detected sequences to the known DNA sequences will help in simplifying the prediction of individual alleles constituting a VDJ recombination.

Acknowledgements. The authors would like to thank Dr. Mary Poss for introducing us to the problem and her constant guidance and extreme patience in explaining the problem. We would also like to thank Bhargavi Panchangam and Dr. Daniel Elleder for insightful discussions.

References

1. Interactive Image Segmentation with Conditional Random Fields, vol. 2 (2008)
2. Boyd, S.D., Marshall, E.L., Merker, J.D., Maniar, J.M., Zhang, L.N., Sahaf, B., Jones, C.D., Simen, B.B., Hanczaruk, B., Nguyen, K.D., Nadeau, K.C., Egholm, M., Miklos, D.B., Zehnder, J.L., Fire, A.Z.: Measurement and clinical monitoring of human lymphocyte clonality by massively parallel v-d-j pyrosequencing. Science Translational Medicine 1(12), 12–23 (2009)
3. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. 16(5), 1190–1208 (1995)
4. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. J. Mach. Learn. Res. 3, 951–991 (2003)
5. Edgar, R.C.: Search and clustering orders of magnitude faster than blast. Bioinformatics 26(19), 2460–2461 (2010)
6. Fippiat, J.-P., Williams, S.C., Tomlinson, L.M., Cook, G.P., Cherif, D., Le Paslier, D., Collins, J.E., Dunham, l., Winter, G., Lefranc, M.-P.: Organization of the human immunoglobulin lambda light-chain locus on chromosome 22q11.2. Human Molecular Genetics 4(6), 983–991 (1995)

7. Gata, B.A., Malming, H.R., Jackson, K.J.L., Bain, M.E., Wilson, P., Collins, A.M.: ihmmune-align: hidden markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences. Bioinformatics 23(13), 1580–1587 (2007)
8. Giudicelli, V., Chaume, D., Lefranc, M.-P.: IMGT/V-QUEST, an integrated software program for immunoglobulin and T cell receptor VJ and VD J rearrangement analysis. Nucleic Acids Research 32(suppl. 2), W435–W440 (2004)
9. Jackson, K.J.L., Boyd, S., Gaëta, B.A., Collins, A.M.: Benchmarking the performance of human antibody gene alignment utilities using a 454 sequence dataset. Bioinformatics 26(24), 3129–3130 (2010)
10. Jung, D., Giallourakis, C., Mostoslavsky, R., Alt, F.W.: Mechanism and control of v(d)j recombination at the immunoglobulin heavy chain locus. Annual Review of Immunology 24(1), 541–570 (2006)
11. Kudo, T.: Crf++: Yet another crf toolkit (2005)
12. Lafferty, J., Mccallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proc. 18th International Conf. on Machine Learning, pp. 282–289. Morgan Kaufmann, San Francisco (2001)
13. Lefranc, M.-P.: Imgt, the international immunogenetics database: a high-quality information system for comparative immunogenetics and immunology. Developmental &; Comparative Immunology 26(8), 697–705 (2002)
14. Li, M.-H., Lin, L., Wang, X.-L., Liu, T.: Protein protein interaction site prediction based on conditional random fields. Bioinformatics 23(5), 597–604 (2007)
15. Lorenz, W., Straubinger, B., Zachau, H.G.: Physical map of the human immunoglobulin k locus and its implications for the mechanisms of vkjk rearrangement. Nucleic Acids Research 15(23), 9667–9676 (1987)
16. Mccallum, A., Li, W.: Early results for named entity recognition with conditional random fields (2003)
17. Munshaw, S., Kepler, T.B.: SoDA2: a Hidden Markov Model approach for identification of immunoglobulin rearrangements. Bioinformatics 26(7), 867–872 (2010)
18. Neuberger, M.S.: Antibody diversification by somatic mutation: from burnet onwards. Immunolo. Cell Biol. 86, 124–132 (2008)
19. Richter, D.C., Ott, F., Auch, A.F., Schmid, R., Huson, D.H.: MetaSim A Sequencing Simulator for Genomics and Metagenomics. PLoS ONE 3(10), e3373+ (2008)
20. Souto-Carneiro, M.M., Longo, N.S., Russ, D.E., Sun, H.-W.W., Lipsky, P.E.: Characterization of the human Ig heavy chain antigen binding complementarity determining region 3 using a newly developed software algorithm, JOINSOLVER.. Journal of immunology (Baltimore, Md.: 1950) 172(11), 6790–6802 (2004)
21. Volpe, J.M., Cowell, L.G., Kepler, T.B.: Soda: implementation of a 3d alignment algorithm for inference of antigen receptor recombinations. Bioinformatics 22(4), 438–444 (2006)