The Maximum-Likelihood Model

The input of a standard phylogenetic analysis consists of a multiple sequence alignment with n taxa and m columns (sites). The output is an *unrooted* binary tree. In order to compute the likelihood on a fixed tree topology, a Markov model comprising several parameters is also needed: the instantaneous nucleotide substitution matrix Q which contains the transition probabilities for time dt between nucleotides, the prior probabilities of observing the nucleotides (i.e., $\pi_A, \pi_C, \pi_G, \pi_T$), which can be determined empirically from the alignment, α (i.e., a parameter that determines the shape of the Γ distribution, which is used to approximate rate-heterogeneity among sites;(Yang, 1994)) of rate heterogeneity, and finally the $2n-3$ edge lengths. The CAT approximation of rate heterogeneity among sites (Stamatakis, 2006) can be used as an efficient and accurate computational approximation to the use of a Γ model, since it requires four times less memory and is three to four times faster than phylogenetic inferences under the Γ model.

In order to compute the likelihood of the data, given the Markov model and a fixed, unrooted, binary tree, one initially needs to compute the internal probability vectors (i.e., the probabilities p(A), p(C), p(G), and p(T) of observing `A, C, G,` or `T` at the internal nodes) for each site. This can be done from the tips of the tree towards a virtual root that can be placed at any edge in the tree. This procedure is also know as Felsenstein (1981) pruning algorithm. Under time-reversibility, the overall likelihood score will be the same regardless of the placement of the virtual root.

Every probability vector entry, $\vec{L}(c)$, at position c ($c = 1...m$) in the alignment and at the tips or internal nodes of the tree topology, contains the four probabilities p(A), p(C), p(G), p(T). The probabilities at the tips of the tree for which observed data *is* available are set to 1.0 (e.g., if the observed nucleotide at site c is A, then $\vec{L}(c) = [1.0, 0.0, 0.0, 0.0]$). Given a parent node k, with two child nodes i and j (with respect to the virtual root), their probability vectors $\vec{L}^{(i)}$ and $\vec{L}^{(j)}$, the respective edge lengths leading to the children b_i and b_j and the transition probability matrices $P(b_i), P(b_j)$, the probability of observing an A at position c of the ancestral sequence, $\vec{L}_A^{(k)}(c)$, is computed as follows:

$$\vec{L}_A^{(k)}(c) = \big(\sum_{S=A}^{T} P_{AS}(b_i)\vec{L}_S^{(i)}(c)\big)\big(\sum_{S=A}^{T} P_{AS}(b_j)\vec{L}_S^{(j)}(c)\big) \tag{1}$$

The transition probability matrix $P(b)$ for a given edge length is obtained from Q by $P(b) = e^{Qb}$. Once the two probability vectors, $\vec{L}^{(i)}$ and $\vec{L}^{(j)}$, to the left and right of the virtual root (vr) have been computed, the likelihood score $l(c)$ for a data site c, can be calculated as follows, given the edge length b_{vr} between nodes i and j:

$$l(c) = \sum_{R=A}^{T} \big(\pi_R \vec{L}_R^{(i)}(c) \sum_{S=A}^{T} P_{RS}(b_{vr})\vec{L}_S^{(j)}(c)\big) \tag{2}$$

The log-likelihood score is then computed by summing over the per-column log-likelihood scores as indicated in Equation 3.

$$LnL = \sum_{c=1}^{m} log(l(c)) \tag{3}$$

In order to maximize Equation 3, all edge lengths, parameters of Q, and α, must be optimized via an ML estimate. For Q and α, the most common approach in state-of-the-art ML implementations consists in using Brent's algorithm (Brent, 1973). For optimization of edge lengths, the Newton-Raphson method is commonly used:the edges are repeatedly visited and their lengths optimized, until the achieved likelihood improvement is smaller than a pre-defined value. Since the edge length is optimized with respect to the likelihood score, the Newton-Raphson method only operates on the likelihood vectors, $\vec{L}^{(i)}$ and $\vec{L}^{(j)}$, that define the edge being optimized. Evidently, when an edge is updated this means that other probability vectors are affected by this change, so they need to be re-computed.

An important implementation issue is the assignment of memory space for the probability vectors. There exist two alternative approaches: a separate vector can be assigned to each of the three outgoing edges of an internal node, or a single vector can be assigned to each internal node. In the latter case, which is significantly more memory-efficient, the probability vectors always maintain a rooted view of the tree (i.e., they are oriented towards the current virtual root of the tree). In the case that the virtual root is then relocated to a different edge (e.g., to optimize the respective edge length), a certain number of vectors, for which the orientation to the virtual root has changed, need to be re-computed. If the tree is traversed in an intelligent way for edge length optimization, the number of probability vectors that will need to be re-computed can be kept to a minimum. RAxML uses this type of rooted probability vector organization.

References

Brent, R. 1973. Algorithms for Minimization without Derivatives. Prentice Hall.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368–376.

Stamatakis, A. 2006. Phylogenetic models of rate heterogeneity: A high performance computing perspective. *in* Proceedings of 20th IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS2006); Rhodos, Greece. 2006.

Yang, Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites. J. Mol. Evol. 39:306–314.

1 Additional Tables

Table 1: Accuracy on randomly sampled short subsequences in terms of normalized edge distance (EDN%) from the original position. The second column shows the average distance of the EPA placements (using *slow* insertions under the $GTR+\Gamma$ model) for the data sets in question. The column EPA-RA shows the average distance for the EPA with previous re-alignment using HMMER. The relative factor compared to EPA is given in parentheses. The last column shows the average ND for a BLAST-based approach, with the relative factors compared to EPA and EPA-RA given in parentheses. All values are shown separately for outer QS and inner QS.

	Data	EPA	EPA-RA	BLAST
all QS	D140	2.02	2.29 (1.14)	3.5 (1.73,1.52)
	D150	0.92	0.98 (1.06)	1.92 (2.09,1.96)
	D218	4.0	4.61 (1.15)	7.28 (1.82,1.58)
	D500	2.34	2.85 (1.22)	4.54 (1.94,1.6)
	D628	2.19	3.02 (1.38)	2.42 (1.1,0.8)
	D714	2.4	2.57 (1.07)	3.61 (1.51,1.4)
	D855	1.82	1.9 (1.04)	2.26 (1.24,1.19)
	D1604	1.1	1.24 (1.13)	1.63 (1.48,1.32)
outer QS	D140	1.88	2.22 (1.18)	2.97 (1.58,1.34)
	D150	0.61	0.67 (1.1)	1.05 (1.72,1.56)
	D218	3.82	4.42 (1.16)	7.2 (1.88,1.63)
	D500	2.33	2.63 (1.13)	4.12 (1.77,1.57)
	D628	2.2	3.06 (1.39)	2.4 (1.09,0.79)
	D714	2.28	2.43 (1.06)	3.04 (1.33,1.25)
	D855	1.71	1.8 (1.05)	2.09 (1.22,1.16)
	D1604	0.93	1.06 (1.14)	1.25 (1.35,1.18)
inner QS	D140	3.29	2.96 (0.9)	8.47 (2.58,2.86)
	D150	2.66	2.71 (1.02)	6.83 (2.56,2.51)
	D218	4.85	5.53 (1.14)	7.69 (1.59,1.39)
	D500	3.12	4.14 (1.33)	7.06 (2.26,1.71)
	D628	2.08	2.65 (1.27)	2.52 (1.21,0.95)
	D714	2.82	3.1 (1.1)	5.75 (2.04,1.85)
	D855	2.47	2.52 (1.02)	3.31 (1.34,1.31)
	D1604	2.06	2.24 (1.08)	3.73 (1.81,1.67)

Table 2: Accuracy of the placement of 2x50 bp paired-end reads. The values given are the node-distance (ND) and the normalized edge distance (NED %). The methods used are the Evolutionary Placement Algorithm (EPA) (*slow* insertions under the $GTR + \Gamma$ model) and BLAST-based nearest neighbor. The relative factor compared to EPA is given in parentheses.

		ND		NED %	
	Data	EPA	BLAST	EPA	BLAST
all QS	D150	3.58	6.67 (1.86)	2.43	5.63 (2.32)
	D218	2.93	6.86 (2.35)	5.96	14.13 (2.37)
	D500	3.74	12.06 (3.23)	6.27	18.72 (2.99)
	D628	1.31	3.56 (2.71)	1.43	3.46 (2.42)
	D714	2.13	4.18 (1.96)	3.2	5.96 (1.86)
	D855	3.82	9.52 (2.49)	2.59	7.46 (2.89)
	D1604	2.49	5.79 (2.32)	1.31	3.94 (3.0)
outer QS	D150	3.84	6.21 (1.62)	1.9	4.65 (2.45)
	D218	2.7	6.35 (2.35)	5.44	13.37 (2.46)
	D500	3.72	12.23 (3.29)	6.09	19.22 (3.16)
	D628	1.26	3.46 (2.74)	1.43	3.63 (2.55)
	D714	2.01	4.1 (2.04)	2.91	5.7 (1.96)
	D855	3.82	9.33 (2.44)	2.55	7.16 (2.8)
	D1604	2.53	5.73 (2.26)	1.21	3.62 (2.99)
inner QS	D150	2.1	9.2 (4.38)	5.37	11.08 (2.07)
	D218	4.0	9.29 (2.32)	8.4	17.72 (2.11)
	D500	3.86	11.03 (2.86)	7.33	15.66 (2.14)
	D628	1.8	4.55 (2.53)	1.5	1.9 (1.27)
	D714	2.61	4.48 (1.72)	4.32	6.93 (1.6)
	D855	3.79	10.69 (2.82)	2.79	9.36 (3.36)
	D1604	2.27	6.13 (2.7)	1.9	5.76 (3.04)

2 Additional Figures

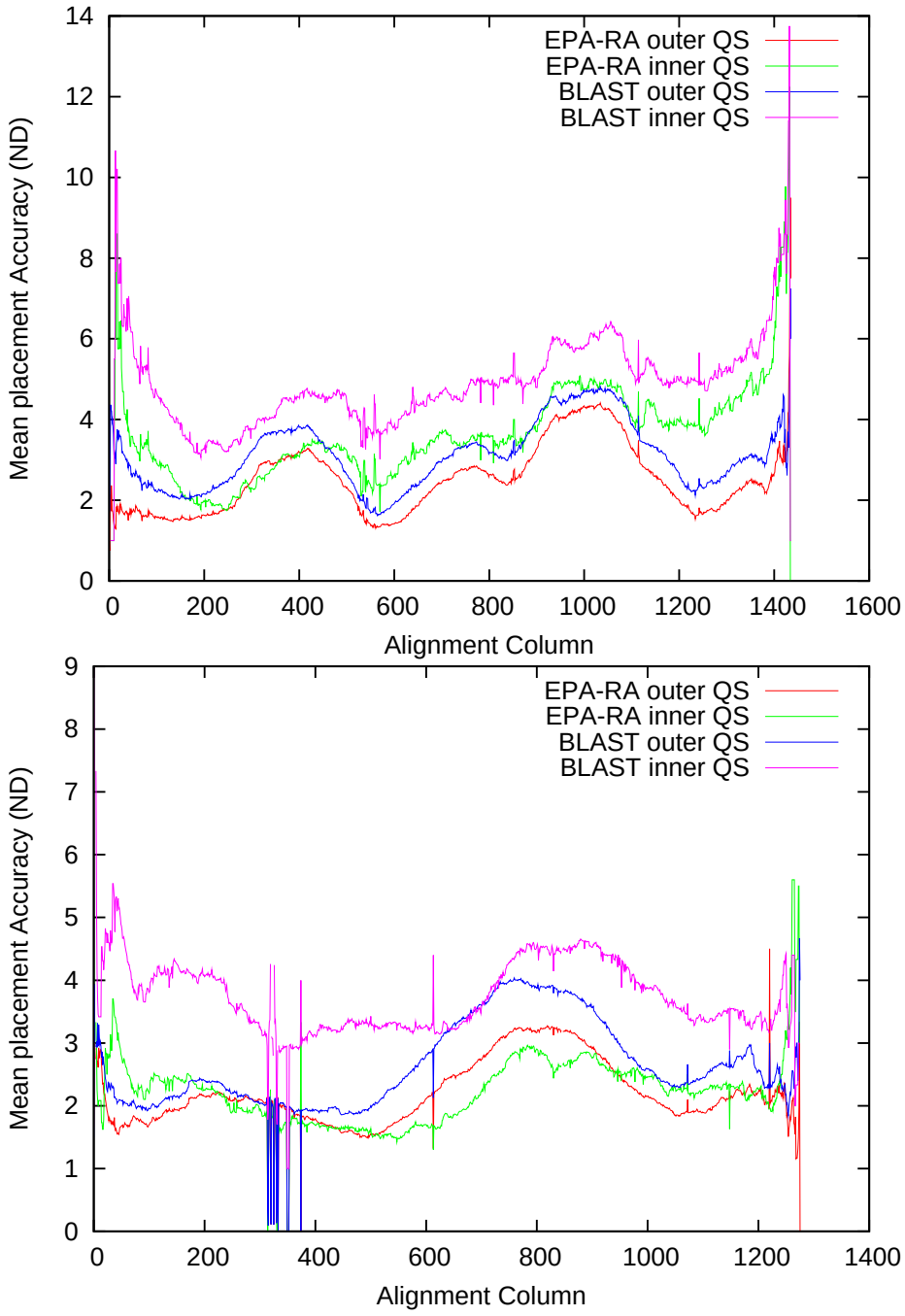

Figure 1: Accuracy profiles for D855 (upper) and D1604 (lower). The plots are derived form the evaluation results of the contiguous sub-sequence placement experiments. For every alignment column the graph shows the mean placement accuracy (ND) over all QS fragments that covered the column. Note that the columns at both ends of the alignment were covered by less fragments than the columns in the middle (due to the uniform sampling of subsequneces fomr the original QS). Therefore the graphs contain a high amount of noise near the extreme ends of the alignment. Similarily there is low coverage and high noise in regions that contain a high amount of gaps (see columns 300 to 400 in D1604). Both graphs show that the EPA has higher placement accuracy than SEQ-NN over the whole lenght of the alignments. Also the EPA shows less decrease in accuracy on the harder sub-set of inner QS.

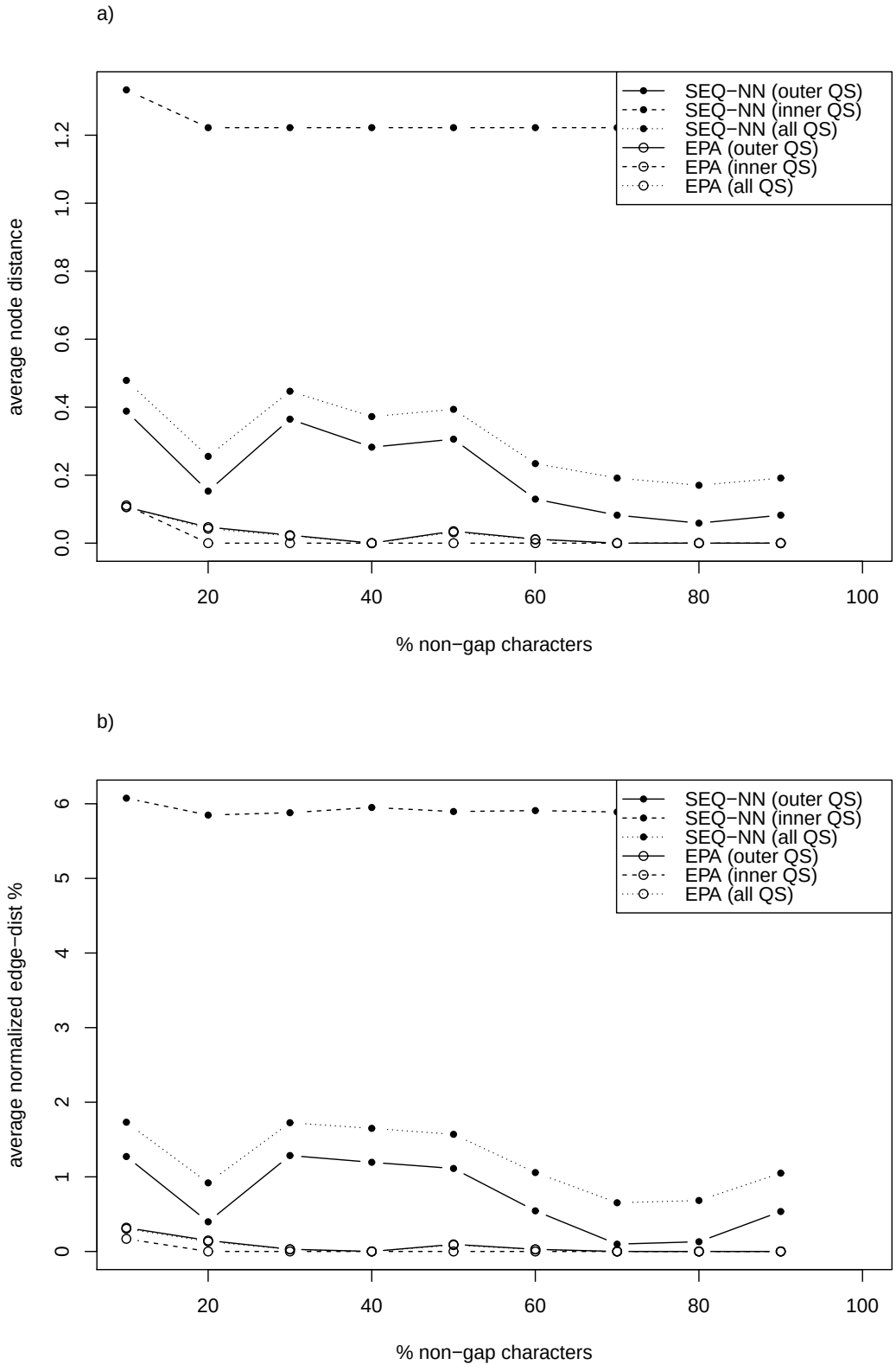

Figure 2: Placement accuracy for QS with artificially introduced random gaps on D140. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

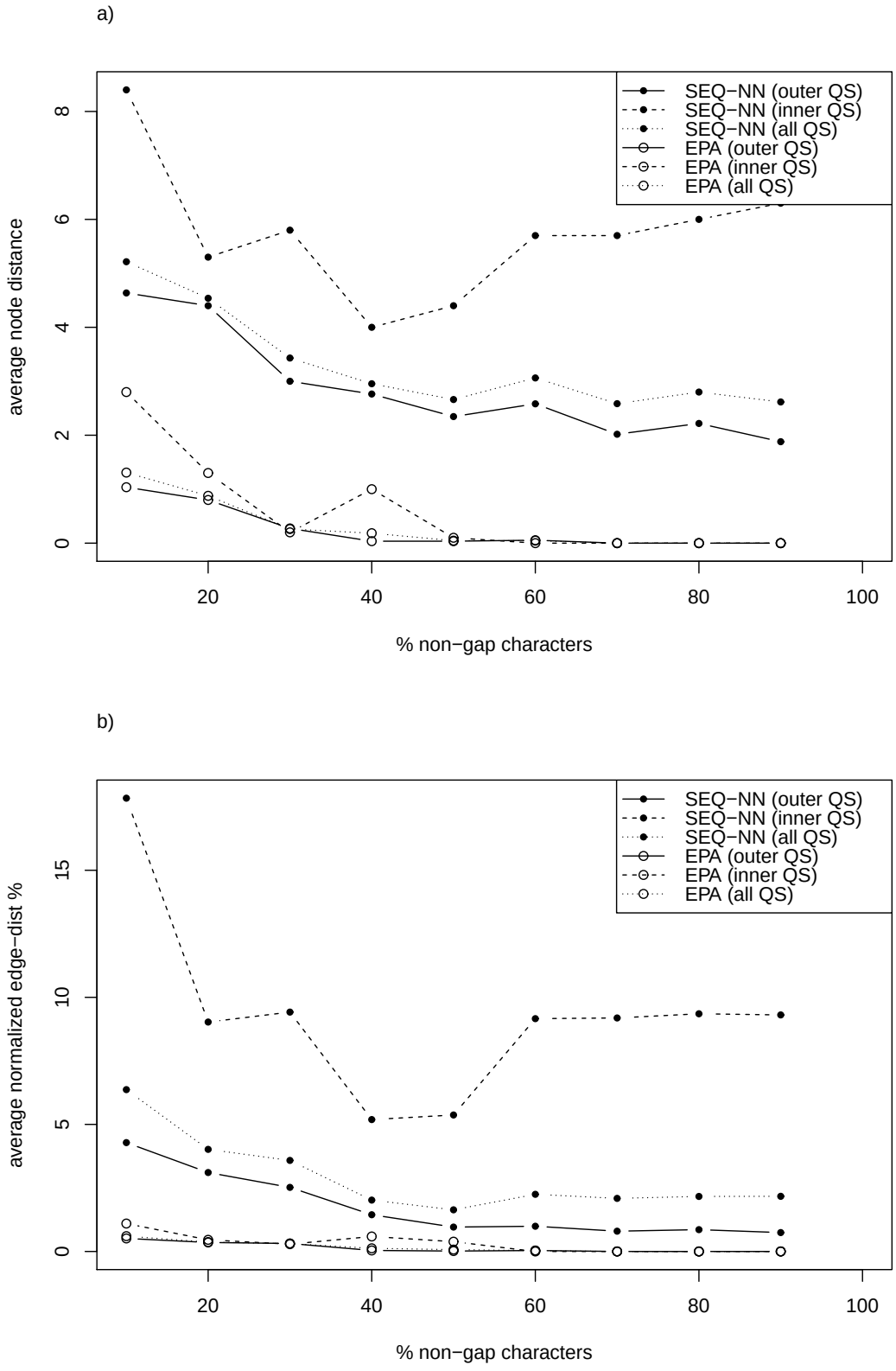

Figure 3: Placement accuracy for QS with artificially introduced random gaps on D150. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

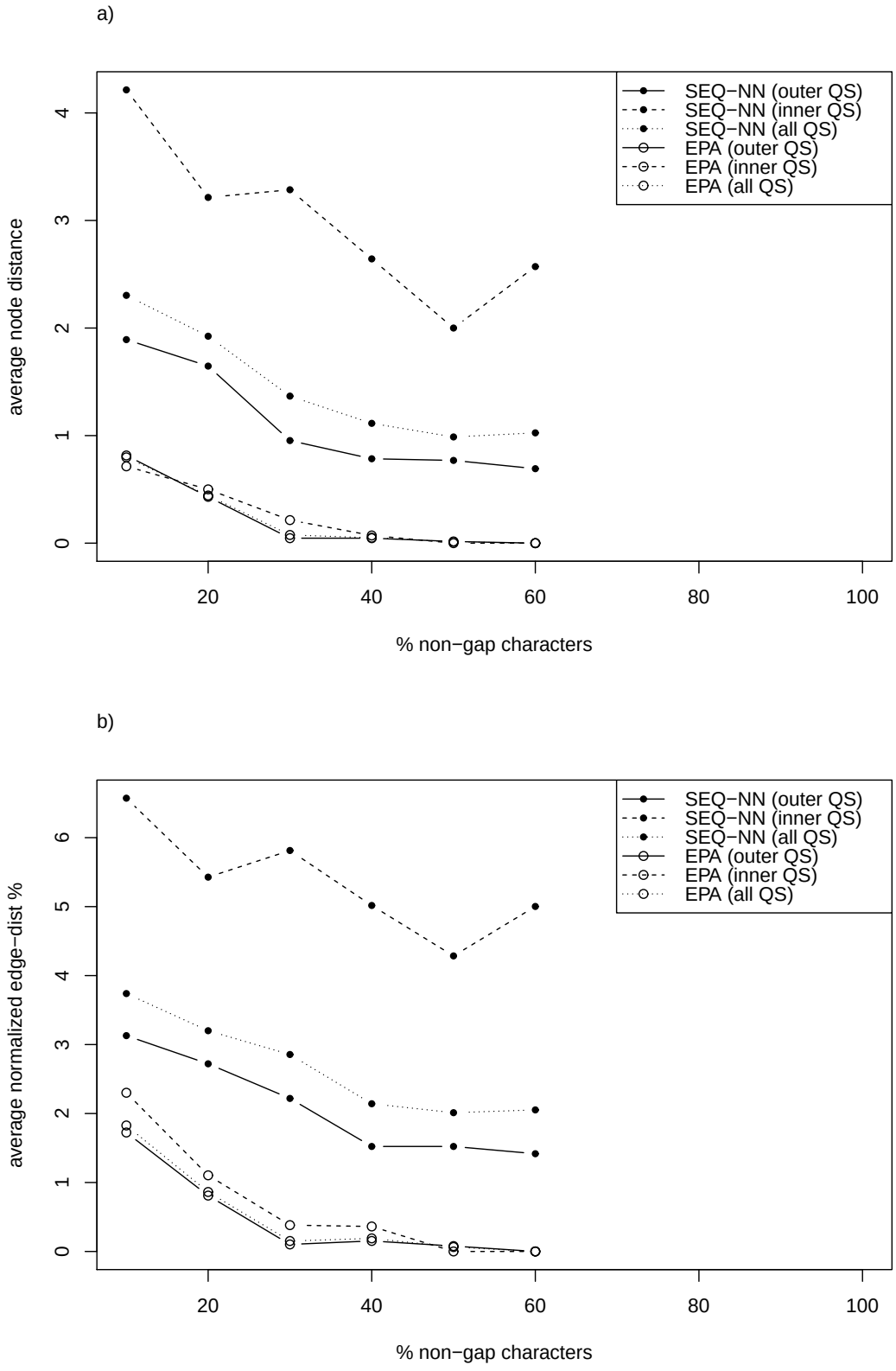

Figure 4: Placement accuracy for QS with artificially introduced random gaps on D218. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

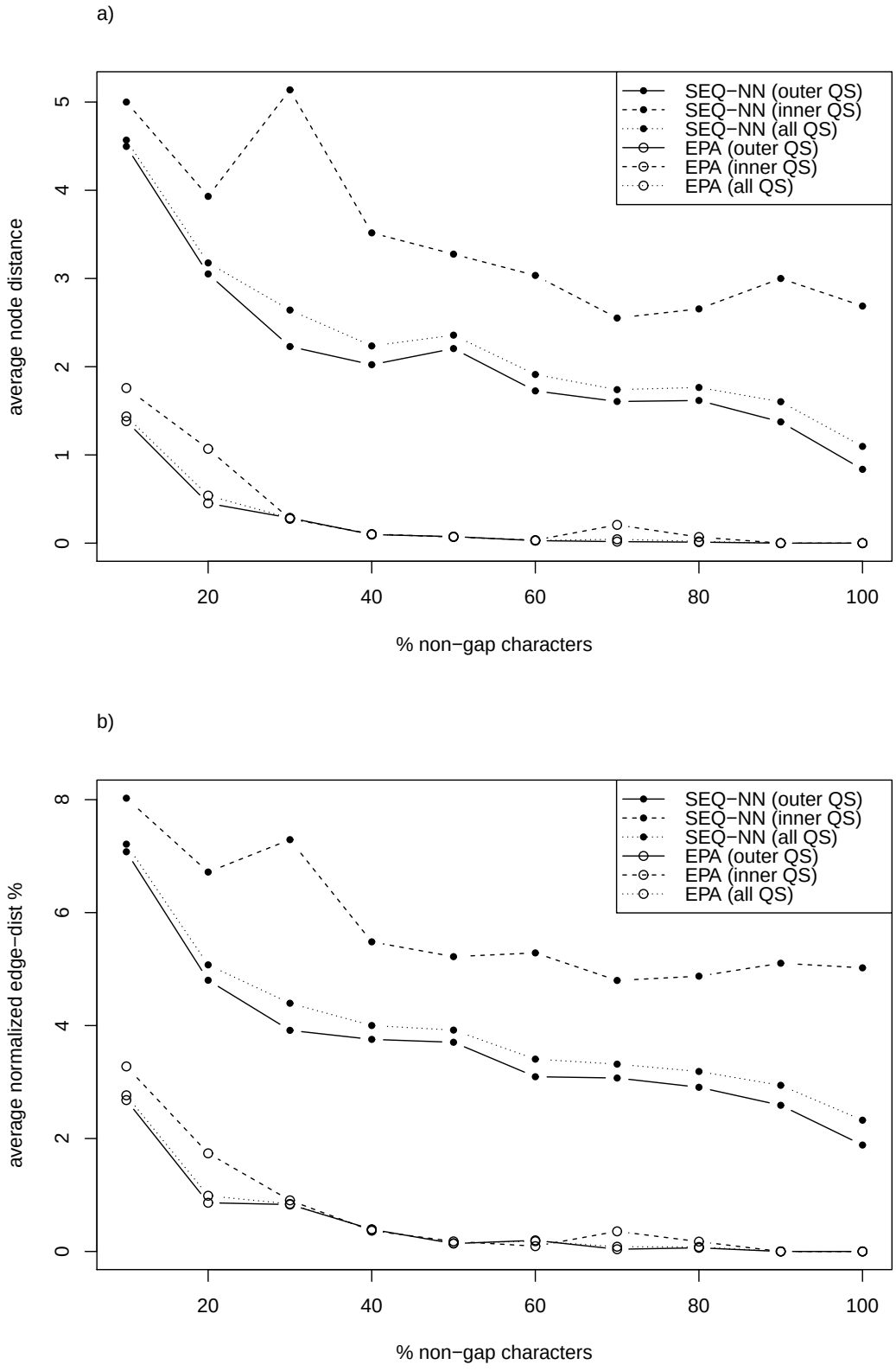

Figure 5: Placement accuracy for QS with artificially introduced random gaps on D500. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

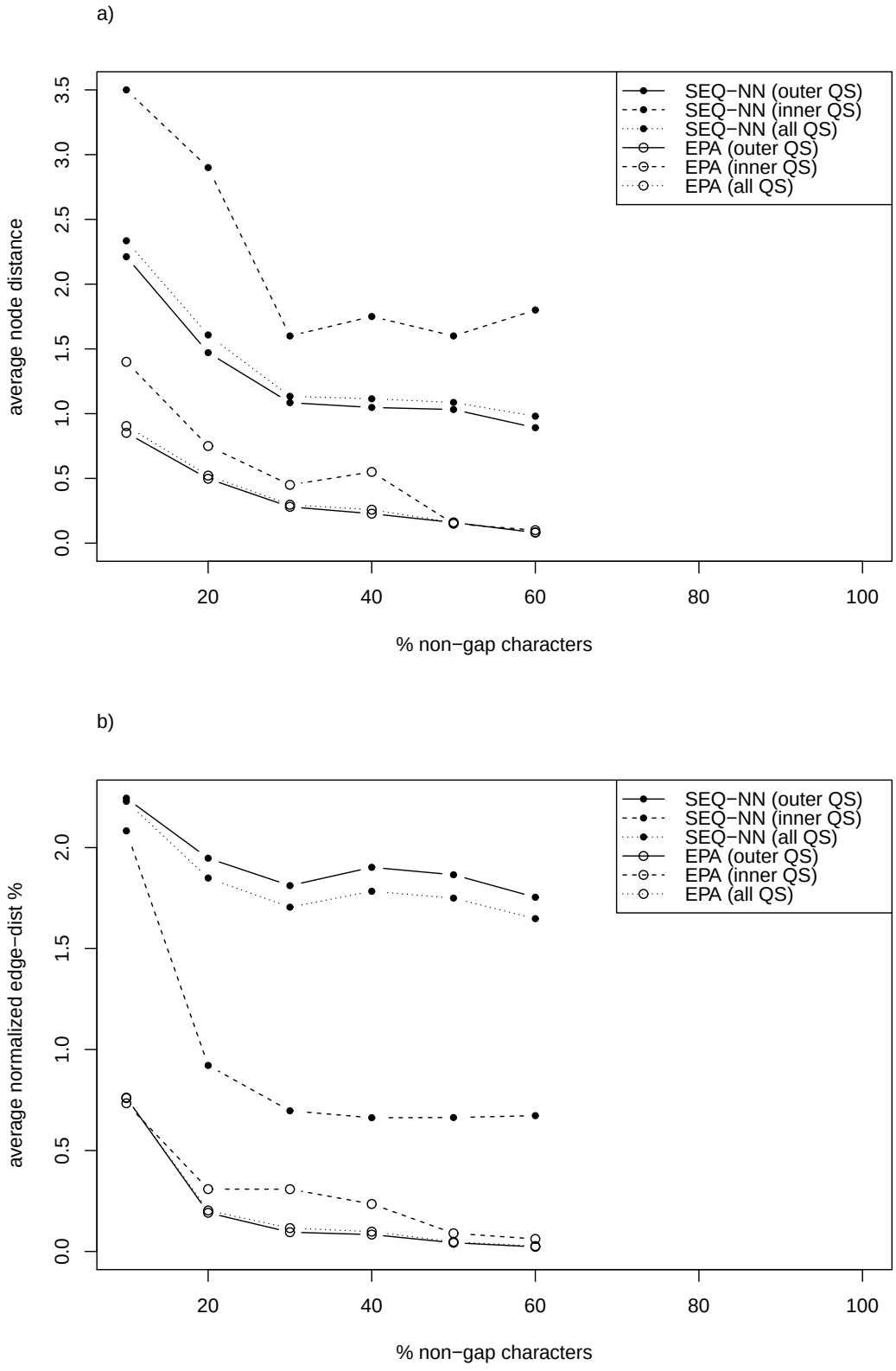

Figure 6: Placement accuracy for QS with artificially introduced random gaps on D628. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

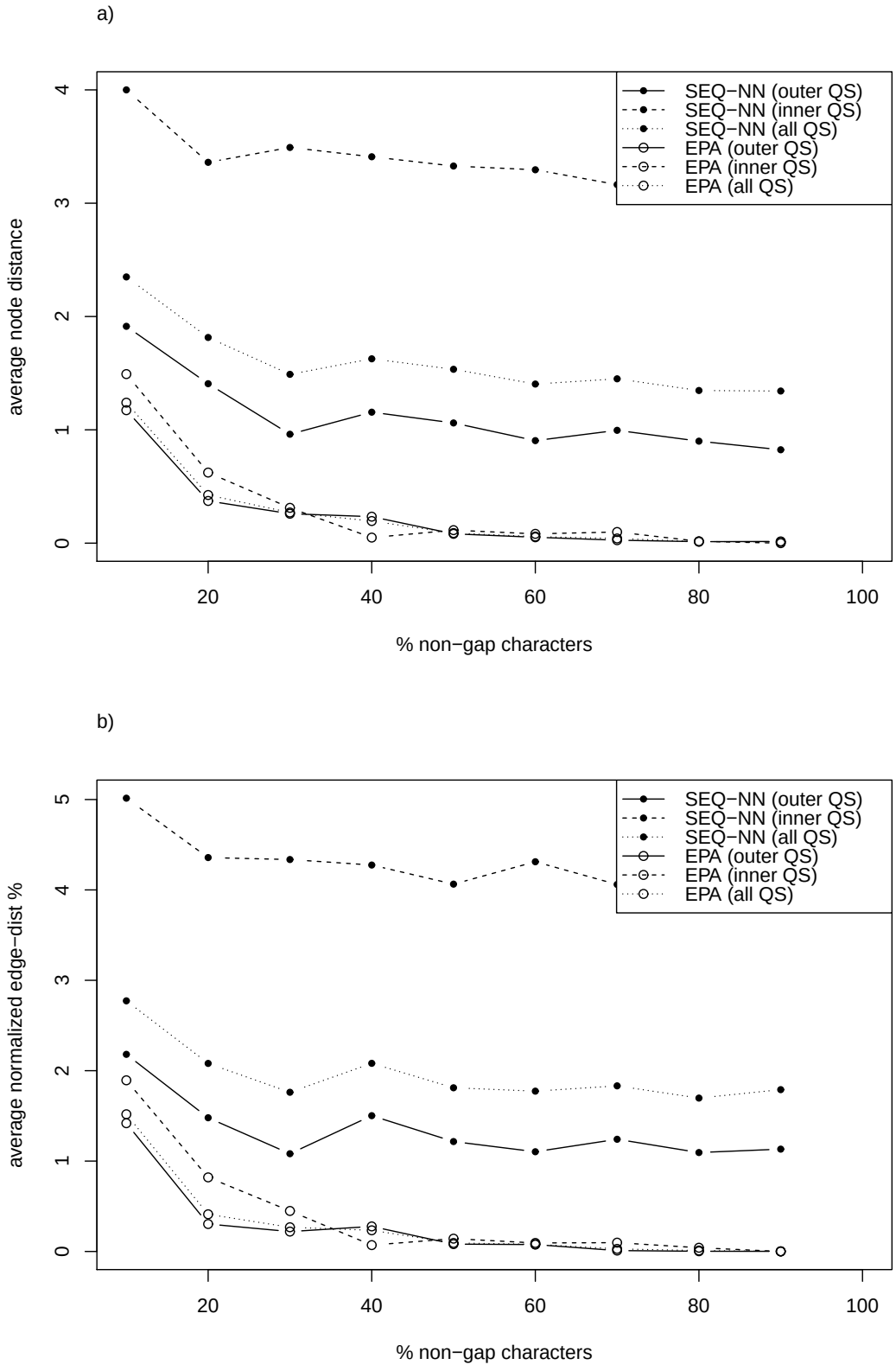

Figure 7: Placement accuracy for QS with artificially introduced random gaps on D714. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

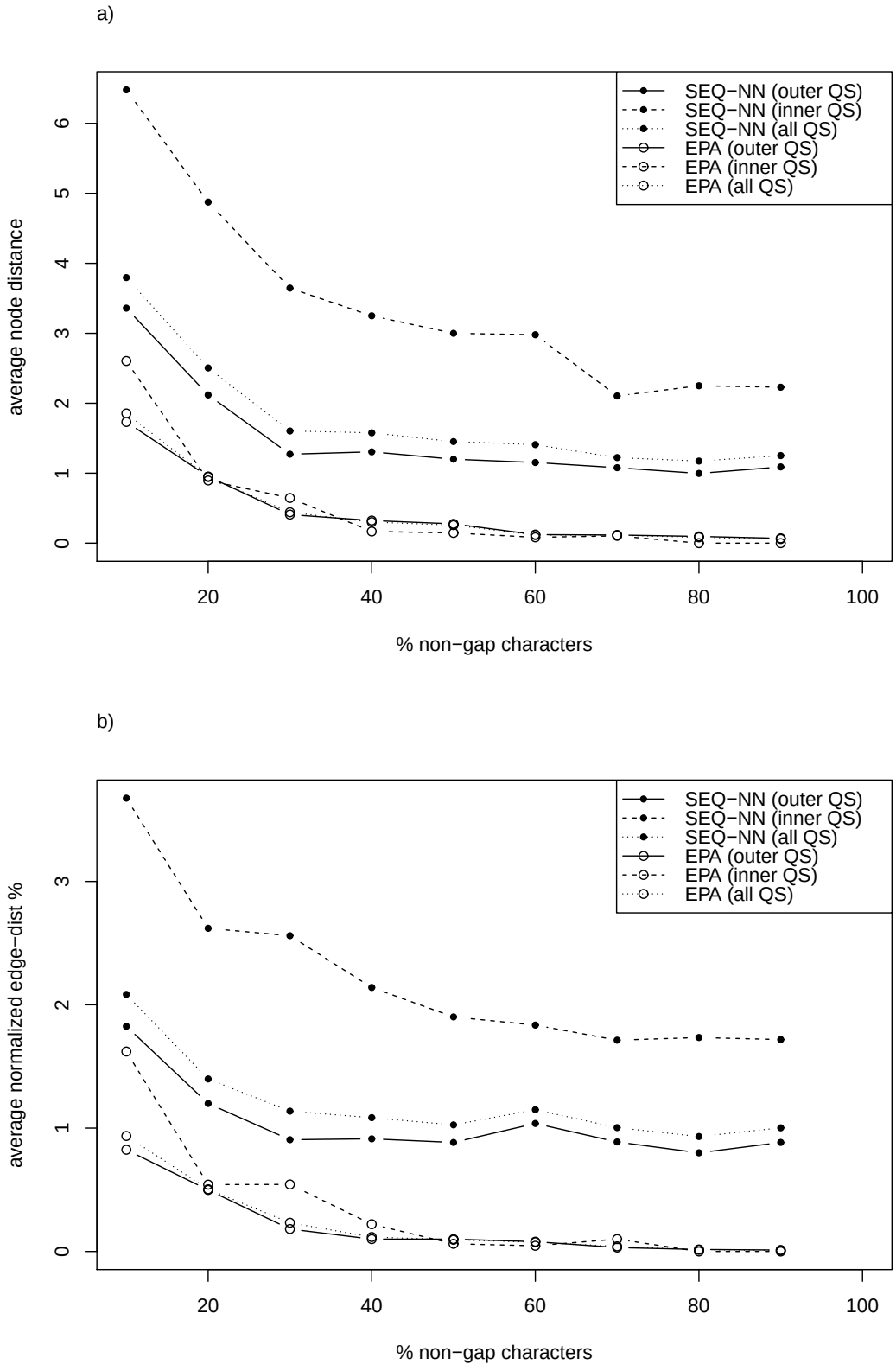

Figure 8: Placement accuracy for QS with artificially introduced random gaps on D855. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

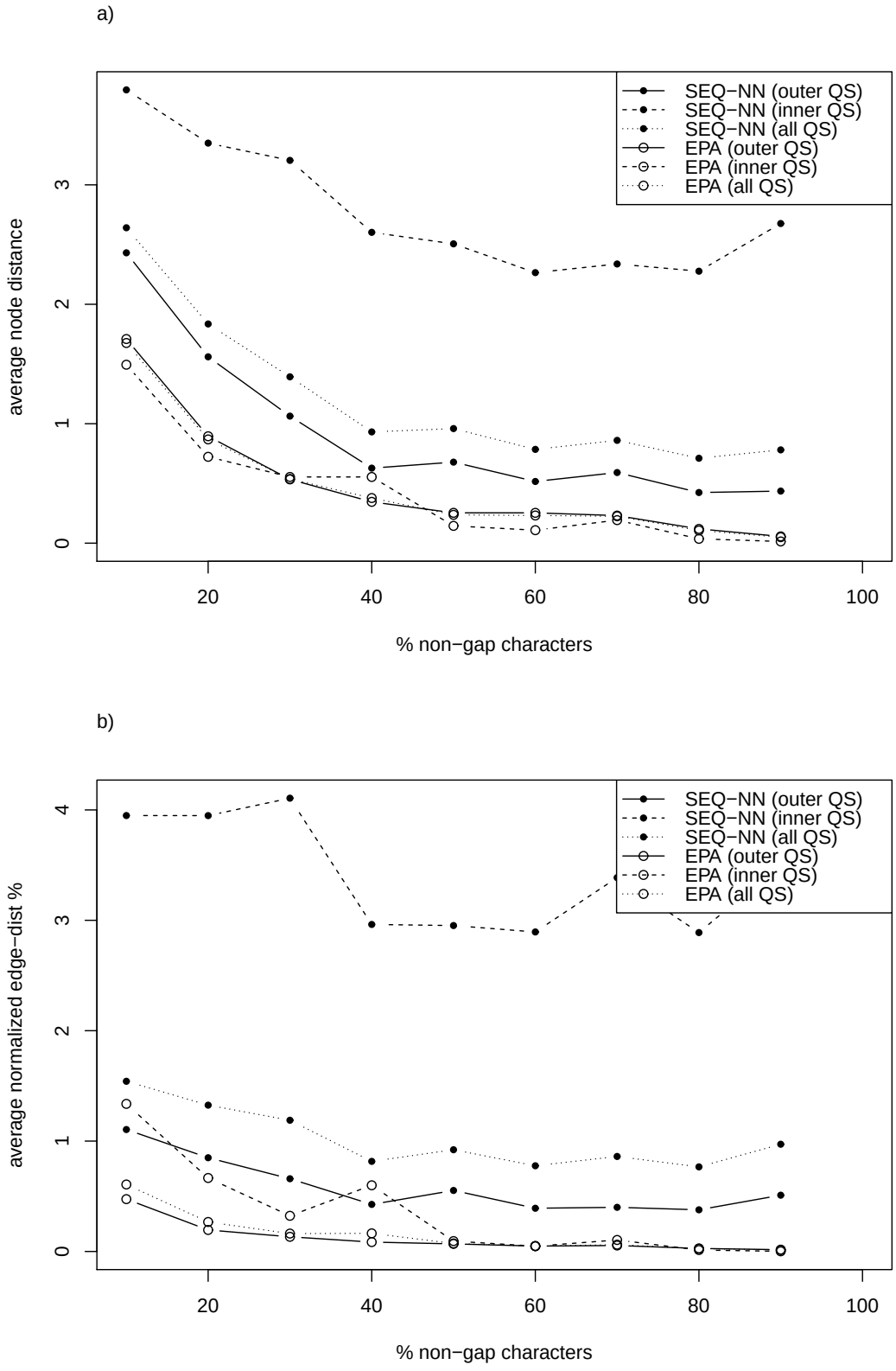

Figure 9: Placement accuracy for QS with artificially introduced random gaps on D1604. (a) Average node distance and (b) normalized edge distance (between insertion positions and real positions).

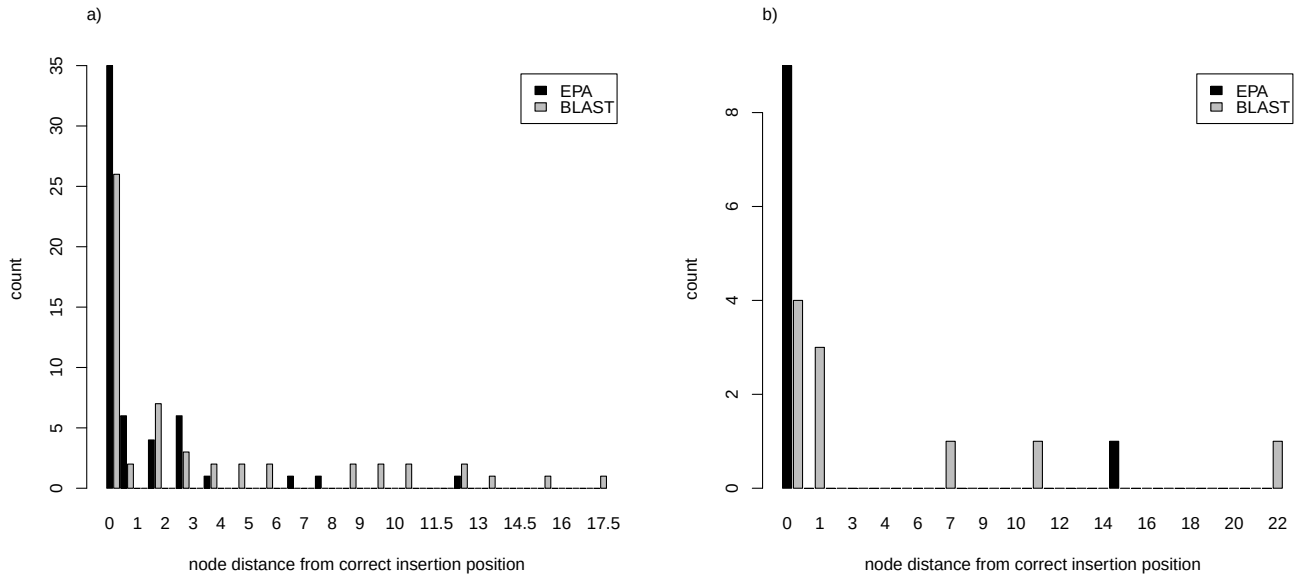

Figure 10: Histogram showing the placement accuracy, based on node distances, for the placement of 2x100 bp paired-end reads from D150, using outer (a) and inner (b) QS.

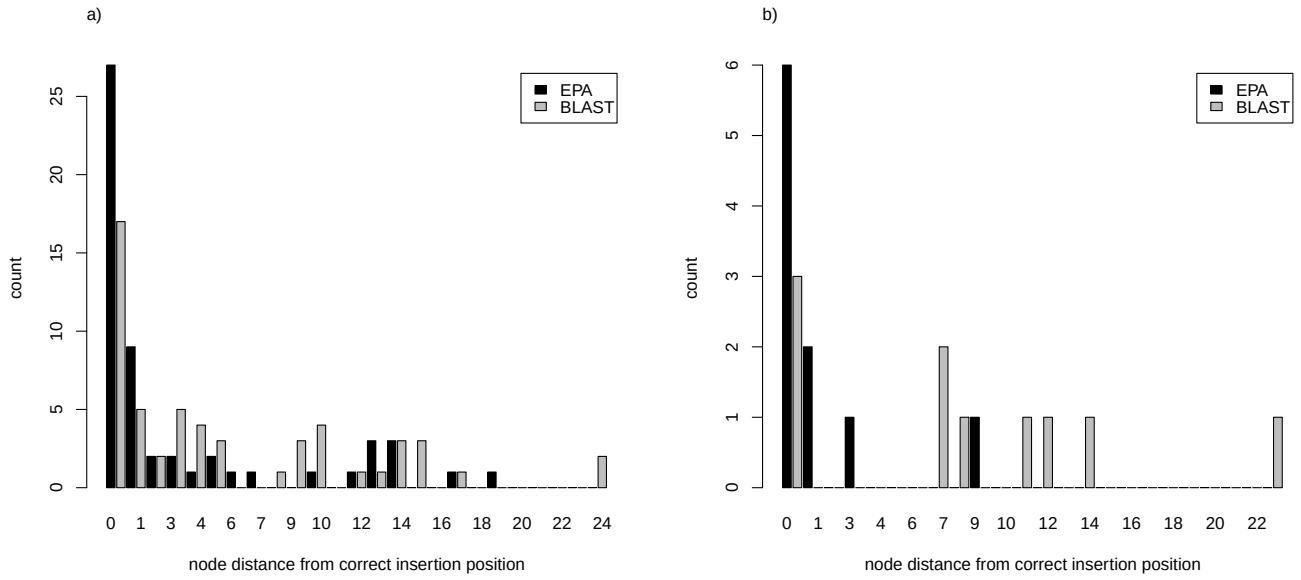

Figure 11: Histogram showing the placement accuracy, based on node distances, for the placement of 2x50 bp paired-end reads from D150, using outer (a) and inner (b) QS.

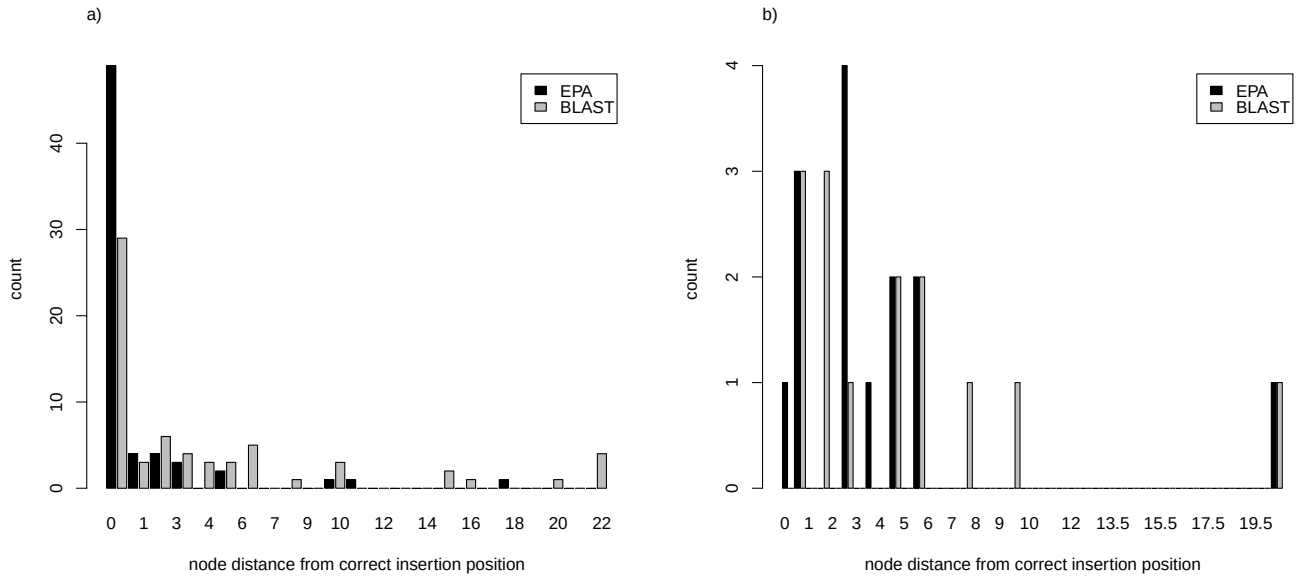

Figure 12: Histogram showing the placement accuracy, based on node distances, for the placement of 2x100 bp paired-end reads from D218, using outer (a) and inner (b) QS.

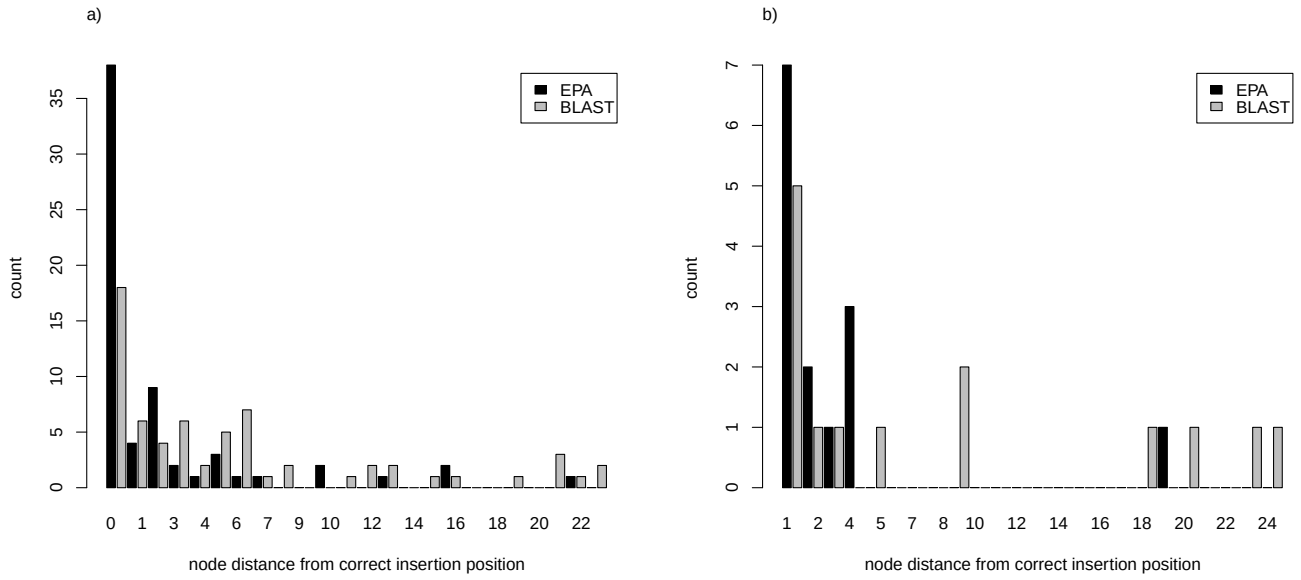

Figure 13: Histogram showing the placement accuracy, based on node distances, for the placement of 2x50 bp paired-end reads from D218, using outer (a) and inner (b) QS.

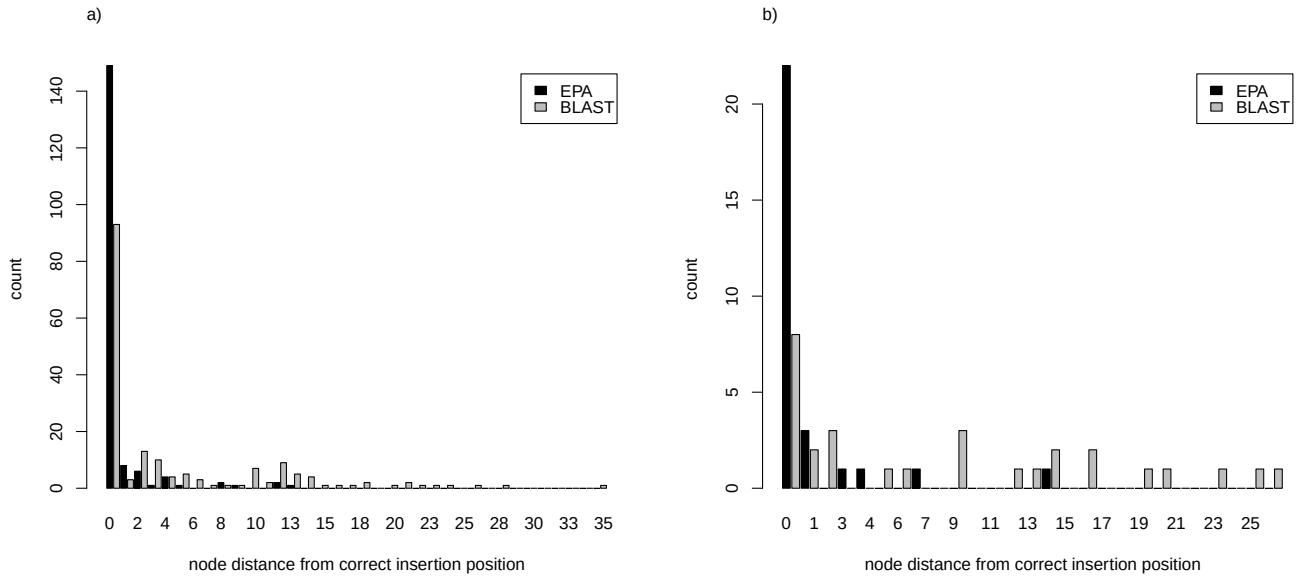

Figure 14: Histogram showing the placement accuracy, based on node distances, for the placement of 2x100 bp paired-end reads from D500, using outer (a) and inner (b) QS.

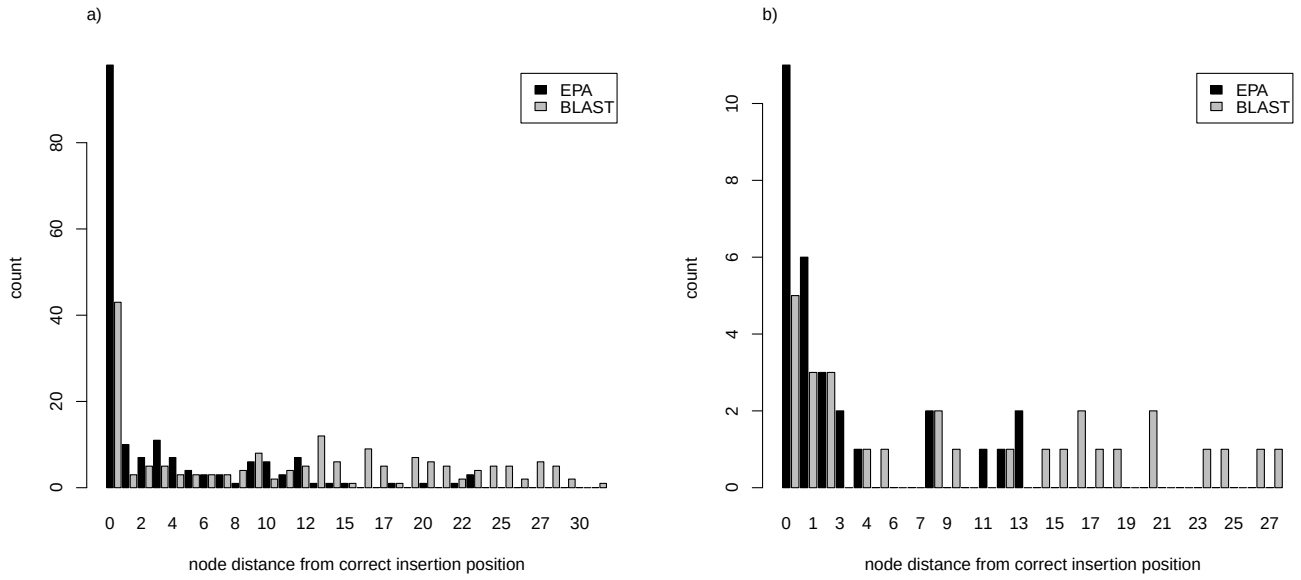

Figure 15: Histogram showing the placement accuracy, based on node distances, for the placement of 2x50 bp paired-end reads from D500, using outer (a) and inner (b) QS.

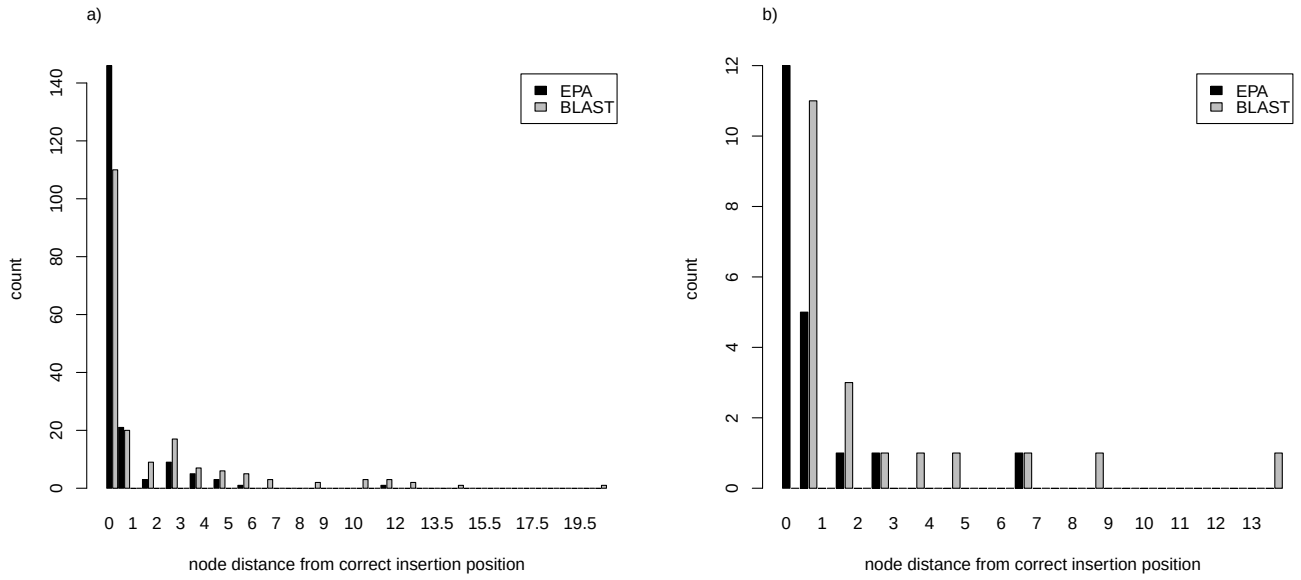

Figure 16: Histogram showing the placement accuracy, based on node distances, for the placement of 2x100 bp paired-end reads from D628, using outer (a) and inner (b) QS.

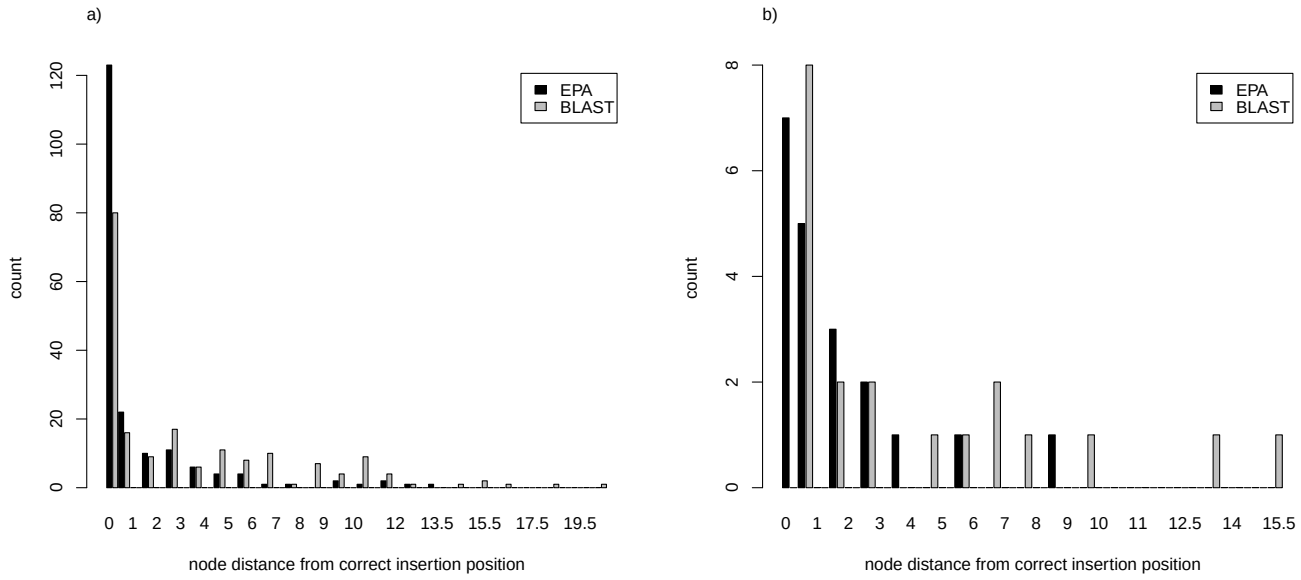

Figure 17: Histogram showing the placement accuracy, based on node distances, for the placement of 2x50 bp paired-end reads from D628, using outer (a) and inner (b) QS.

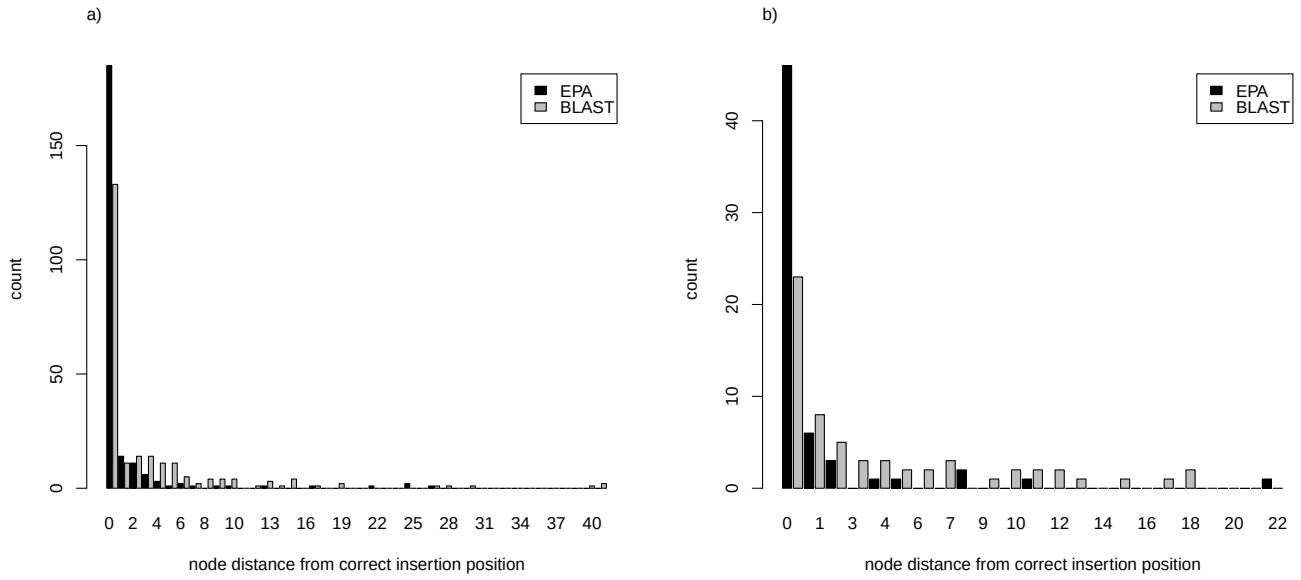

Figure 18: Histogram showing the placement accuracy, based on node distances, for the placement of 2x100 bp paired-end reads from D714, using outer (a) and inner (b) QS.

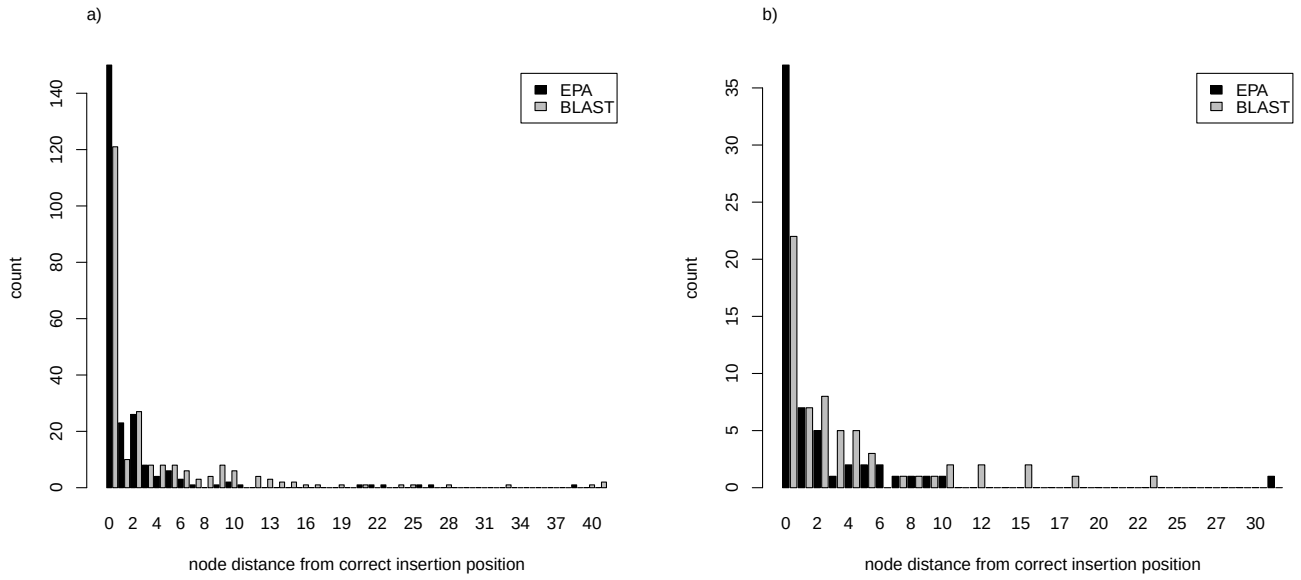

Figure 19: Histogram showing the placement accuracy, based on node distances, for the placement of 2x50 bp paired-end reads from D714, using outer (a) and inner (b) QS.

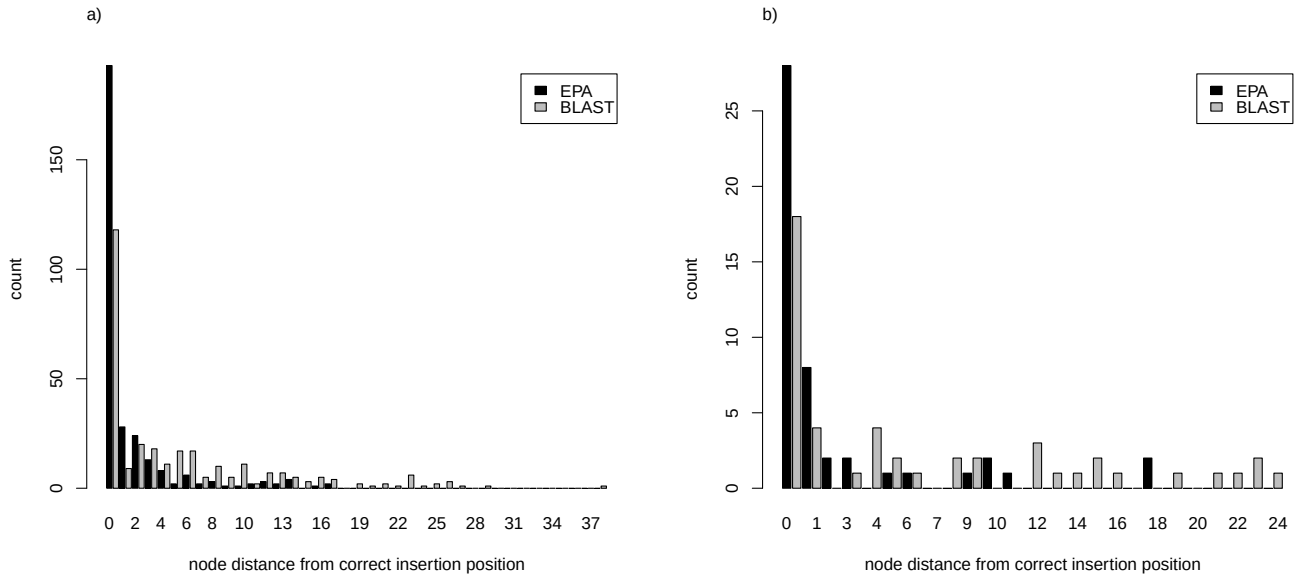

Figure 20: Histogram showing the placement accuracy, based on node distances, for the placement of 2x100 bp paired-end reads from D855, using outer (a) and inner (b) QS.

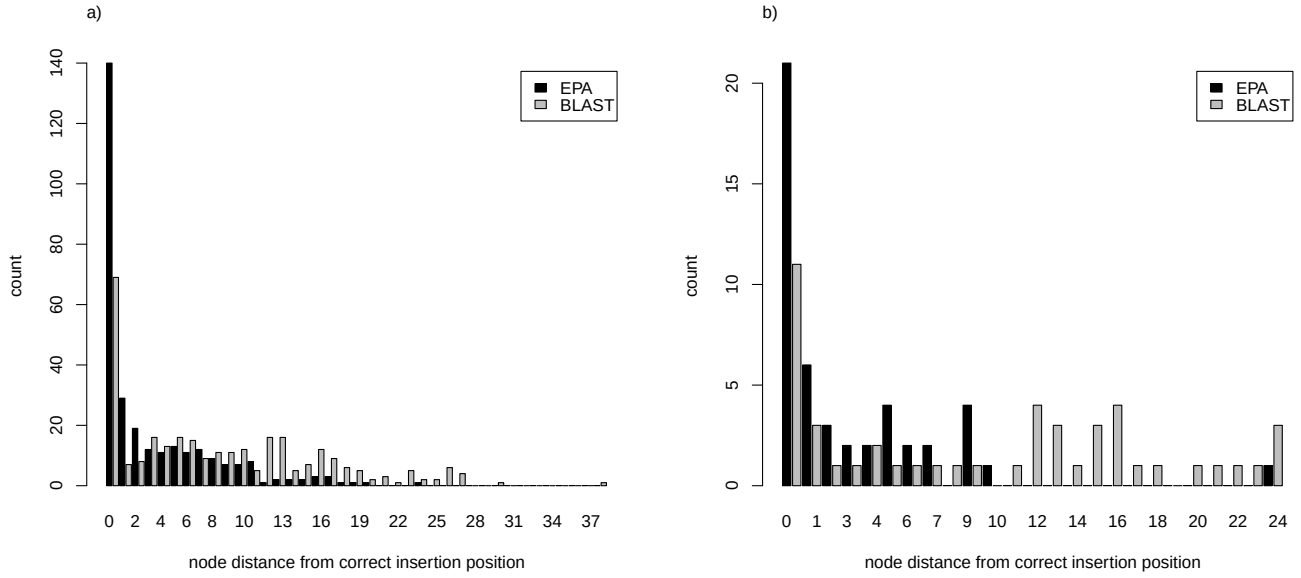

Figure 21: Histogram showing the placement accuracy, based on node distances, for the placement of 2x50 bp paired-end reads from D855, using outer (a) and inner (b) QS.

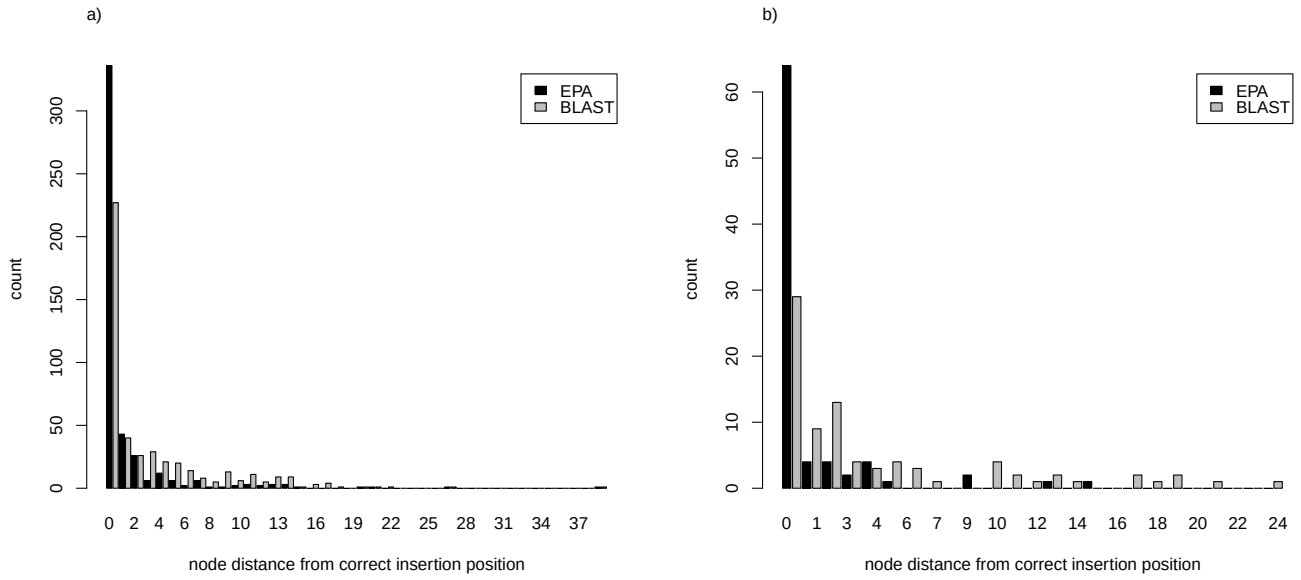

Figure 22: Histogram showing the placement accuracy, based on node distances, for the placement of 2x100 bp paired-end reads from D1604, using outer (a) and inner (b) QS.

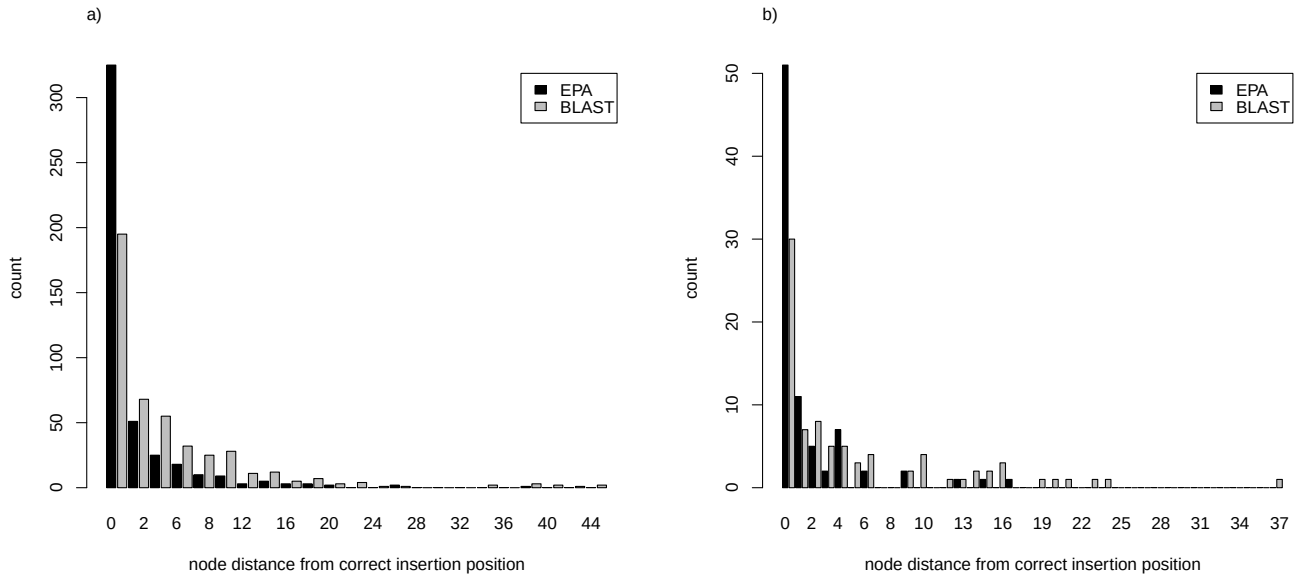

Figure 23: Histogram showing the placement accuracy, based on node distances, for the placement of 2x50 bp paired-end reads from D1604, using outer (a) and inner (b) QS.

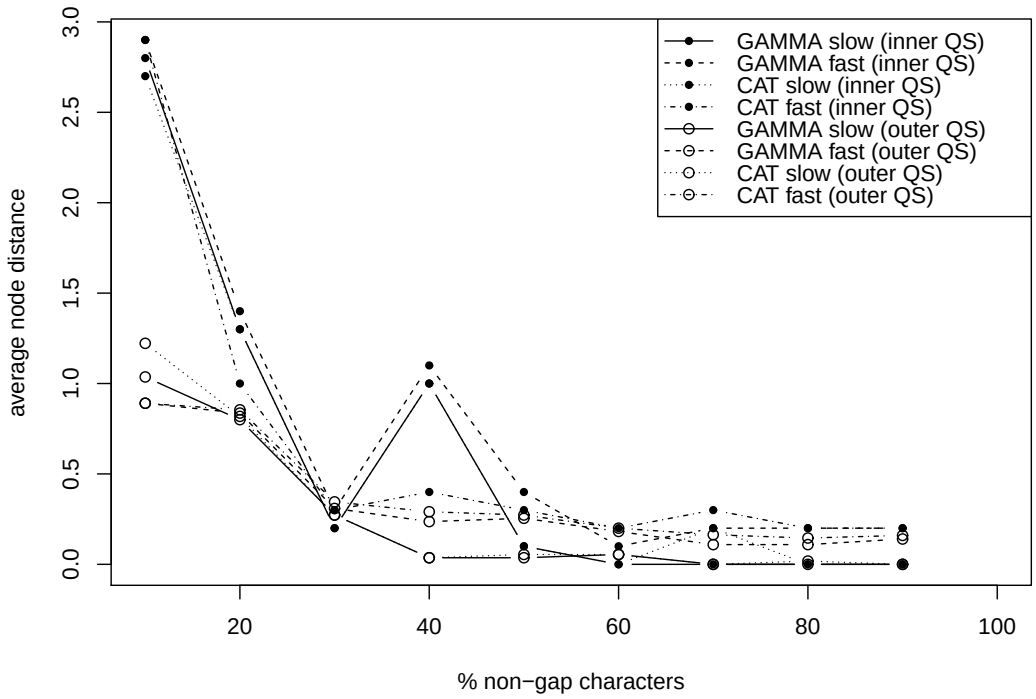

Figure 24: Average Node Distance between insertion positions and real positions for different versions of the EPA (*fast*/*slow* insertions) algorithm and model types (GTR+Γ, GTR+CAT) on all QS from D150.

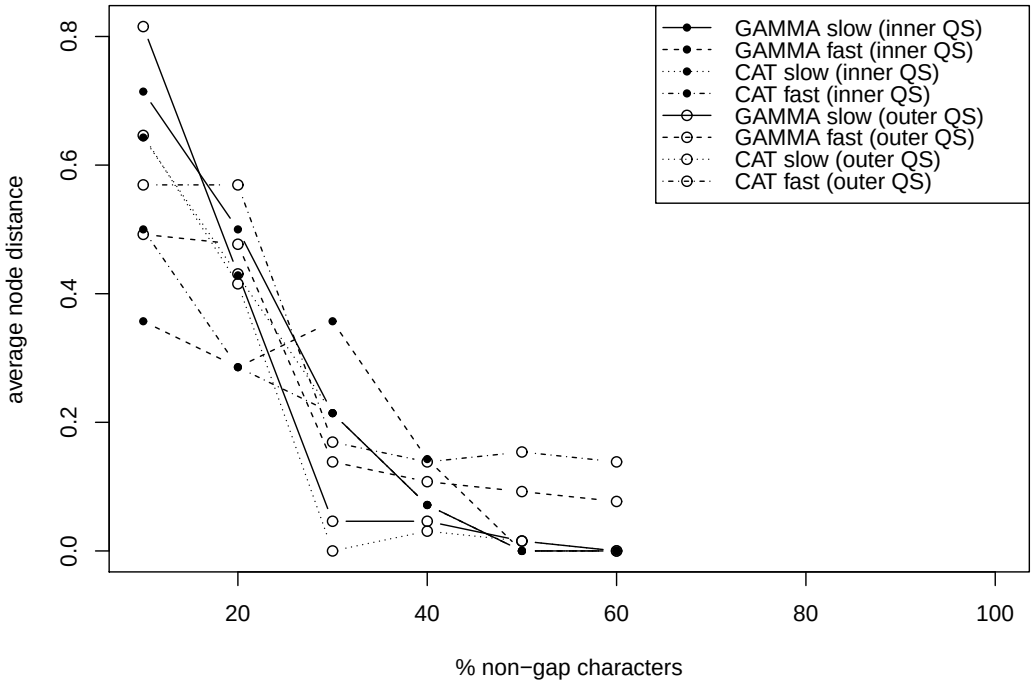

Figure 25: Average Node Distance between insertion positions and real positions for different versions of the EPA (*fast*/*slow* insertions) algorithm and model types (GTR+Γ, GTR+CAT) on all QS from D218.

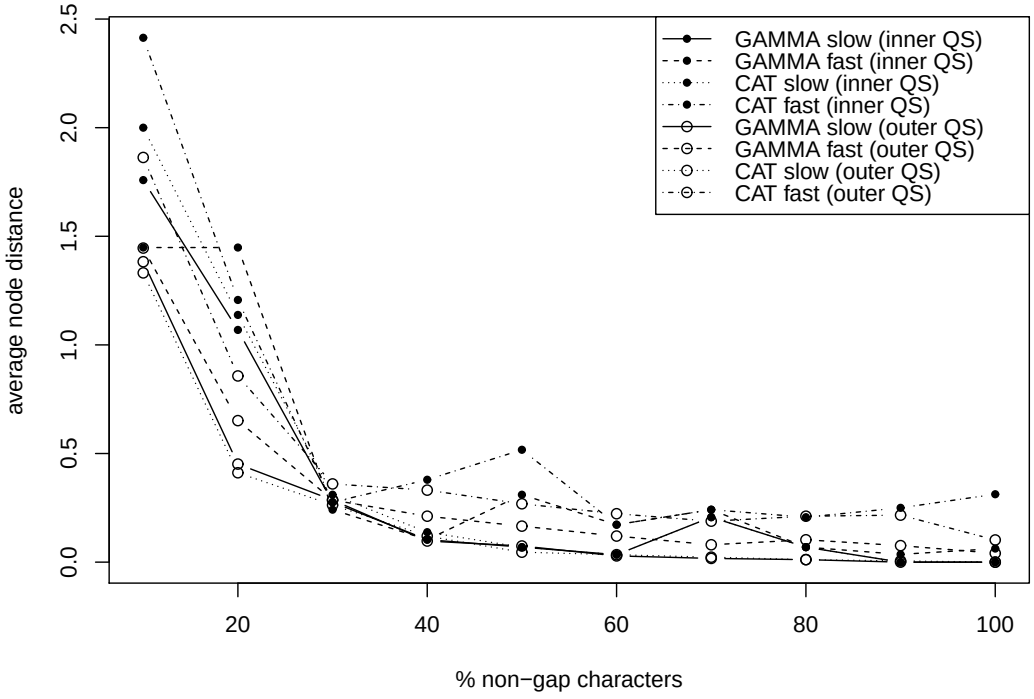

Figure 26: Average Node Distance between insertion positions and real positions for different versions of the EPA (*fast*/*slow* insertions) algorithm and model types (GTR+Γ, GTR+CAT) on all QS from D500.

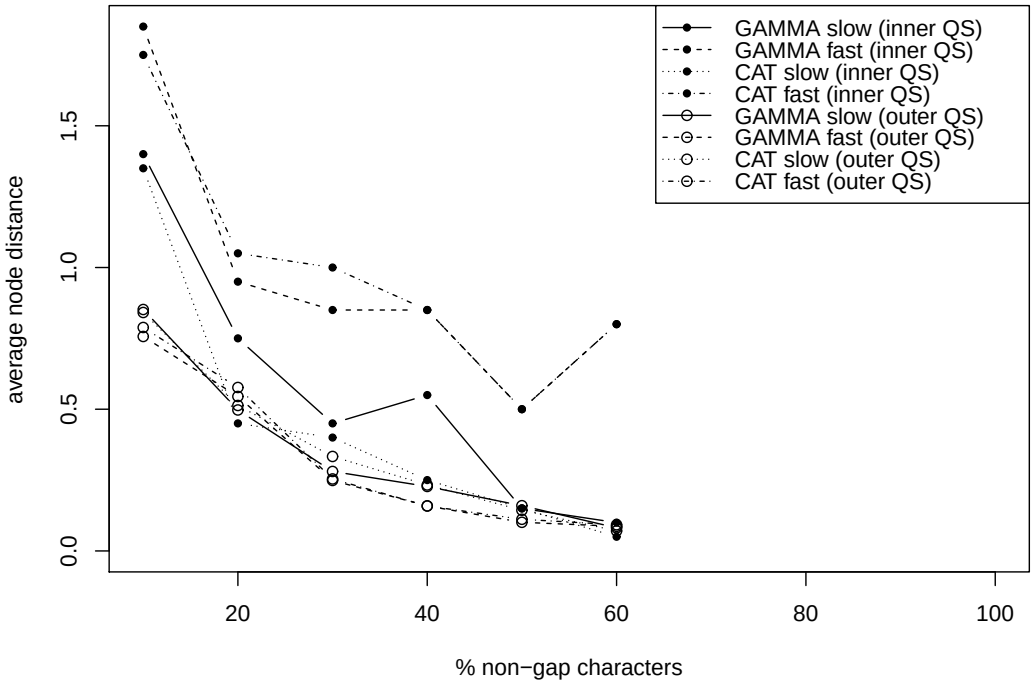

Figure 27: Average Node Distance between insertion positions and real positions for different versions of the EPA (*fast*/*slow* insertions) algorithm and model types (GTR+Γ, GTR+CAT) on all QS from D628.

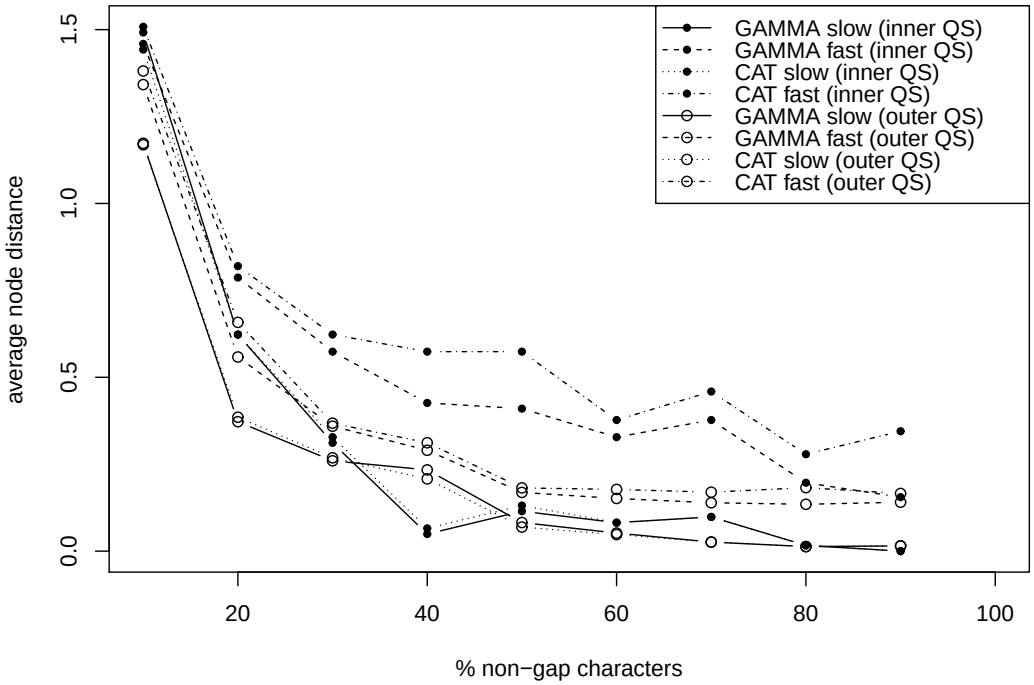

Figure 28: Average Node Distance between insertion positions and real positions for different versions of the EPA (*fast*/*slow* insertions) algorithm and model types (GTR+Γ, GTR+CAT) on all QS from D714.

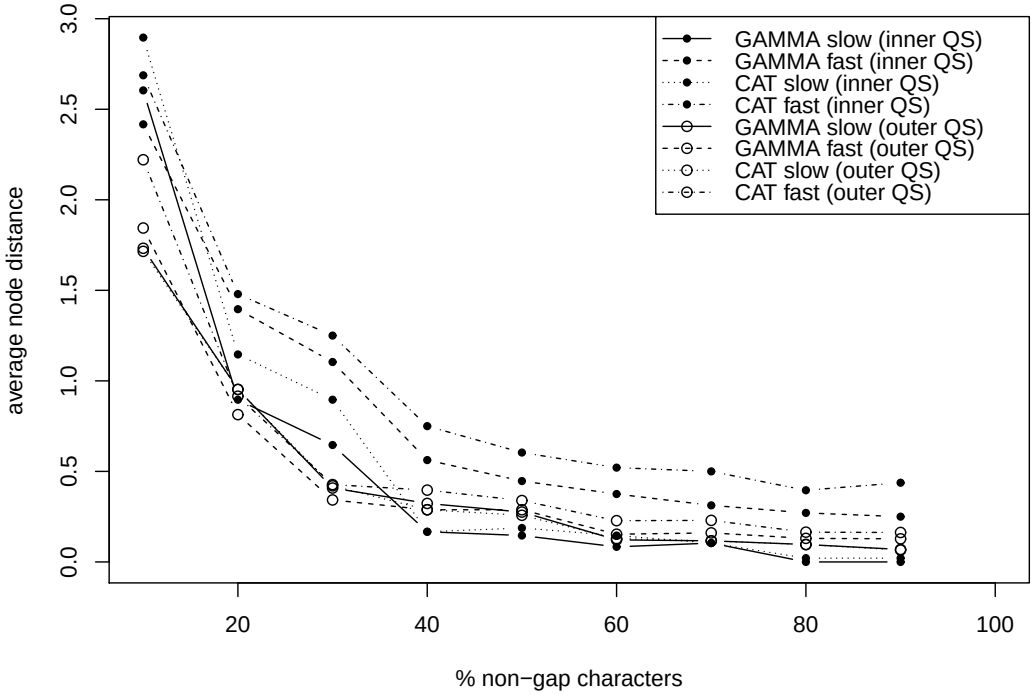

Figure 29: Average Node Distance between insertion positions and real positions for different versions of the EPA (*fast*/*slow* insertions) algorithm and model types (GTR+Γ, GTR+CAT) on all QS from D855.

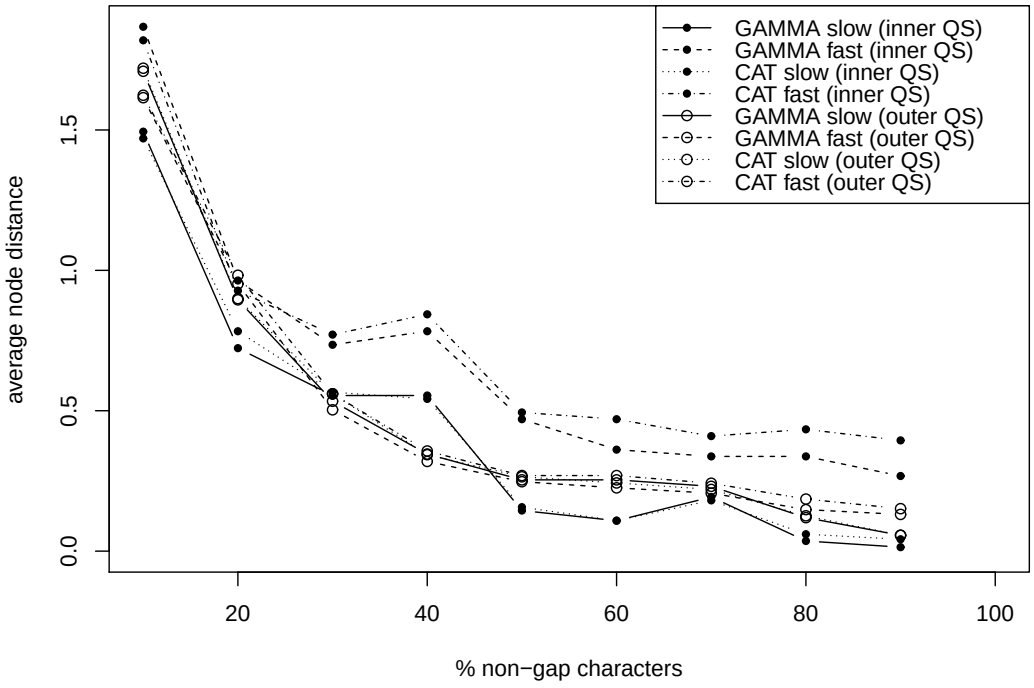

Figure 30: Average Node Distance between insertion positions and real positions for different versions of the EPA (*fast*/*slow* insertions) algorithm and model types (GTR+Γ, GTR+CAT) on all QS from D1604.