A Likelihood Function

Here we shall derive the likelihood function $P(D|M)$ of the model described in Section 2 of the manuscript. In short, it gives the probability that a set of data points is produced by a particular model. For the sake of generality, the model is defined in a two dimensional space (X-Y). The formalism that is developed here can be directly applied to the analysis of the T-L data by relabeling X as time and Y as length.

Let us start by considering one data point and one line segment. Assuming that measurements of both coordinates are independently affected by some noise with a Gaussian distribution, we can write the probability that a data point $\left(x^k, y^k\right)$ is represented by an infinitesimal part of a line segment in the following form:

$$dP^k \sim \exp\left\{-\frac{1}{2\sigma_x^2}\left(x-x^k\right)^2 - \frac{1}{2\sigma_y^2}\left(y-y^k\right)^2\right\} dl. \tag{2}$$

Here x and y are the coordinates along the segment, dl represents the length of the part of the line segment, σ_x^2 and σ_y^2 are the variances of the corresponding noise distributions.

For the sake of convenience we are going to use parametric equations to represent a line segment. Assume that the coordinates of the starting point of the segment are x_0 and y_0 and the coordinates of the end point are x_0' and y_0'. Let us define two coefficients Δ_x and Δ_y in the following manner:

$$\begin{aligned} \Delta_x &= x_0' - x_0, \\ \Delta_y &= y_0' - y_0. \end{aligned} \tag{3}$$

Now we can express the equation of a line segment in terms of a dimensionless variable ξ:

$$\begin{aligned} x &= x_0 + \Delta_x \xi, \\ y &= y_0 + \Delta_y \xi, \quad 0 \le \xi \le 1. \end{aligned} \tag{4}$$

Using this definition we can express the probability that a data point is part of a particular line segment as an integral over ξ.

$$P^k = A\sqrt{\Delta_x^2 + \Delta_y^2} \int_0^1 \exp\left\{-\frac{1}{2\sigma_x^2}\left(x_0 + \Delta_x\xi - x^k\right)^2 - \frac{1}{2\sigma_y^2}\left(y_0 + \Delta_y\xi - y^k\right)^2\right\} d\xi. \tag{5}$$

Here A is a normalization coefficient that will be defined later.

The integral above can be expressed in terms of error functions.

$$P^k = A\frac{\sqrt{\pi}\sqrt{\Delta_x^2+\Delta_y^2}}{2a}\Big[erfc(b)-erfc(a+b)\Big]\times \exp\left\{-\frac{1}{2\sigma_x^2}\left(x_0-x^k\right)^2-\frac{1}{2\sigma_y^2}\left(y_0-y^k\right)^2+b^2\right\}. \quad (6)$$

In order to simplify the notation in the formula presented above we have introduced two values a and b, which are defined as follows.

$$a = \sqrt{\frac{\Delta_x^2}{2\sigma_x^2}+\frac{\Delta_y^2}{2\sigma_y^2}}, \quad (7)$$

$$b = \frac{1}{a}\left(\frac{\Delta_x}{2\sigma_x^2}(x_0-x^k)+\frac{\Delta_y}{2\sigma_y^2}(y_0-y^k)\right). \quad (8)$$

At this point we have derived the expression for the probability that one data point is part of a single line segment. It is easy to extend this formalism to the case of multiple data points and line segments. Assuming that all data points are independent, we can write down an expression for the probability for a collection of n data points being represented by a single line segment as a product of the probabilities for individual data points.

$$P_t = P^1\cdot P^2\cdot ..\cdot P^n. \quad (9)$$

In the case of several line segments, every segment will contribute some probability to each data point, so the probability that a data point k is a part of one segment from a collection of m line segments is given by a sum of probabilities for each segment.

$$P^k = P_1^k+P_2^k+..+P_m^k. \quad (10)$$

Combining Equations 9 and 10 we obtain the general expression for the likelihood function of our model.

$$P_t = \prod_{k=1}^{n}\left(\sum_{i=1}^{m}P_i^k\right). \quad (11)$$

Here the index k is used to label different data points and the index i labels different line segments.

At this point the definition of the likelihood function is still incomplete. We still have to determine the normalization coefficient A. This is done by using the following normalization condition:

$$\frac{A}{L}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left[\sum_{i=1}^{m}P_i^k\right]dx^k dy^k = 1, \quad (12)$$

Here L is the total length of the segmented line. The integrals in the expression Eq. 12 can be easily computed analytically to produce the following value of the normalization coefficient:

$$A = \frac{1}{2\pi\sigma_x\sigma_y}. \tag{13}$$

Now we combine expressions Eq. 6, Eq. 11, and Eq. 13 to produce the final formula for computing the likelihood. Since using very small numbers in a numerical calculation can result in a loss of precision, it is better to use the negative logarithm of the probability instead of the probability itself. So the optimization routine has to minimize the negative logarithm of the posterior probability, which is equivalent to maximizing the value of the posterior probability.

$$\begin{aligned} -\log P_t &= n\log\left(4\sqrt{\pi}\sigma_x\sigma_y\right) - \sum_{k=1}^{n}\log\left(\sum_{i=1}^{m}\frac{1}{a_{ik}}\sqrt{\Delta_{xi}^2+\Delta_{yi}^2}\left[erfc(b_{ik}) - erfc(a_{ik}+b_{ik})\right]\times\right. \\ &\left.\exp\left\{-\frac{1}{2\sigma_x^2}\left(x_{0i}-x^k\right)^2 - \frac{1}{2\sigma_y^2}\left(y_{0i}-y^k\right)^2 + b_{ik}^2\right\}\right). \end{aligned} \tag{14}$$

As before, the index k is used here to label the data points while the index i labels the line segments. The quantities a_{ik} and b_{ik} are defined in the same way as a and b in equations Eq. 7 and Eq. 8 with the segment parameters x_0, y_0, Δ_x, and Δ_y replaced by the corresponding indexed values x_{0i}, y_{0i}, Δ_{xi}, and Δ_{yi}.

Eq. 14 gives us the general expression of the likelihood that a set of data points is represented by a set of connected line segments with added Gaussian noise. As a part of Bayes' theorem Eq. 1 this expression provides us with a measure of the quality of the fit of the model, which is used to find the optimal locations and velocities of the line segments.

B Computer Simulation of Tracking Data

We have developed a computer simulation that generates artificial tracks with known underlying motion overlapped by noise. In this simulation the motion of the cargo is modeled by a linear combination of deterministic motion of the motor complex and the thermal motion of the cargo.

Let us begin by considering the thermal motion of the cargo. It is modeled as biased Brownian motion in two dimensions using the following procedure. First, the instantaneous displacements of the cargo due to the thermal fluctuations, δx and δy, in the X and Y directions are generated independently using random numbers drawn from a Gaussian distribution with zero mean and variance $2D\delta t$. Here D is the effective diffusion coefficient of the cargo and δt is the time step of the simulation. After that, these two independent displacements, in X and Y directions, are combined into a vector $\delta\vec{l}$. Next another vector $\vec{r}$ that specifies the direction and the distance from

the motor to the center of the cargo is determined and the displacement vector $\delta\vec{l}$ is decomposed into a displacement along this direction $\delta\vec{l}_{\parallel}$ and a perpendicular displacement $\delta\vec{l}_{\perp}$. The component of the displacement parallel to $\vec{r}$ is modified to ensure that the cargo always stays within a certain radius from the motor. So, if the parallel component of the cargo displacement is directed toward the motor, it is left unchanged. If the cargo displacement is away from the motor, the magnitude of $\delta\vec{l}_{\parallel}$ is modified in the following way:

$$|\delta\vec{l'}_{\parallel}| = |\delta\vec{l}_{\parallel}| \left(\frac{r_{max} - |\vec{r}|}{r_{max}} \right)^{\frac{1}{n}}. \tag{15}$$

Here r_{max} is the maximum allowed distance from the motor to the cargo. The parameter n is used to change the distribution of the distances between the motor and the cargo. Finally, the total biased displacement of the cargo $\delta\vec{l'}$, caused by the thermal fluctuations, is computed as a sum of $\delta\vec{l'}_{\parallel}$ and $\delta\vec{l}_{\perp}$.

As mentioned above, the total motion of the cargo is a sum of the motion produced by the thermal fluctuations and the deterministic motion of the motor complex. The behavior of the motor complex is determined by a set of states. Every state has a characteristic velocity and length distribution, so when a motor complex is in some state, its velocity and the amount of time that it spends in that state are determined by the properties of the state. So, at every time step the displacement of the motor complex is computed as $\delta t\,\vec{v}_m$, where v_m is the velocity of the motor complex in the current state, and the displacement of the cargo due to fluctuations $\delta\vec{l'}$ is computed as described above. After that the positions of the motor complex $\vec{\xi}_m$ and the cargo $\vec{\xi}_c$ are updated.

$$\begin{aligned} \vec{\xi}_m &= \vec{\xi}_m + \delta t\,\vec{v}_m, \\ \vec{\xi}_c &= \vec{\xi}_c + \delta t\,\vec{v}_m + \delta\vec{l'}. \end{aligned}$$

Finally, a check is performed to determine if the current state should be terminated. If so, a new state is selected. The velocity and the duration of the new state are generated from the distributions associated with the state.

Switching between the states is modeled as a Markov process. There is a transition probability associated with each pair of states. This probability describes the likelihood that the motor complex will transition into the second state in the pair just after in leaves the first state. For example, let us consider a three state model with states that correspond to motion in the the plus-end direction, motion in the minus-end direction, and a pause. Assume that the motor complex is in the plus-end state. When this state terminates the motor complex switches to another state. Since there are three states there are also three transition probabilities. The ”plus-minus” probability, the ”plus-pause” probability, and the ”plus-plus” probability corresponding to the transitions from the plus-end state to the minus-end state, pause, or back to the plus-end state. If we choose the ”plus-plus”

probability to be zero and the remaining to to be equal to 50%, then there is a 50% chance that the motor complex will switch to either a pause state or a minus-end motion state after a plus-end state terminates.

C Outline of the fitting procedure

1. Load X-Y-T data.
2. Obtain L-T data.
   - 2a. Find microtubule location by fitting the X-Y data to a straight line.
   - 2b. Find the projections of the data points along the microtubule. Let L be the position along the microtubule. The L coordinate for the first data point is set to zero. For all other data points it is equal to the distance along the microtubule from the first data point.
3. Initialize the fitting procedure.
   - 3a. Start with one segment which has the first and the last data points of the tracking series as its end points.
   - 3b. Compute perpendicular distances from all data points to the segmented line (initially consisting of only one segment). Find the data point with the largest distance. Use it as a vertex to increase the number of segments by one by splitting the segment that is associated with that data point.
   - 3c. Repeat the procedure described above until the number of segments reaches N_i. The value of N_i is chosen by hand. Usually it is chosen so that initially there is, on average, one segment for a few data points, e.g., one segment for every three data points. In the extreme case the N_i can be chosen so that there is one segment for each data point.
   - 3d. Use the optimization algorithm described in Appendix D: to find positions of the vertices (end points of the segments) that produce the highest posterior probability for N_i segments.
4. Find the optimal fit.
   - 4a. Reduce the number of segments by one.
     - Find a pair of segments that is a good candidate for merging using the procedure outlined in the text.
     - Merge the two segments by removing their common vertex.

- Optimize the positions of remaining vertices to get the highest posterior probability for the model with the number of segments reduced by one.

4b. Continue reducing the number of segments by repeating the procedure described above until the stopping condition is met.

- When using the model selection method based on the calibration procedure, the stopping condition occurs when the negative logarithm of the posterior probability of the model becomes significantly larger than the calibration value for the average duration of a segment. Note that the average duration of a segment is the duration of the track divided by the current number of segments.
- In other cases the stopping condition occurs when reducing the number of segments by one causes the maximum posterior probability of the model to decrease. Here the prior probability has to be chosen in such a way as to make sure that this happens for a finite number of segments.

4c. Backtrack by one step (increase the number of segments by one) to find the model that is the best fit for the tracking data. This model is the final answer to the problem we are solving.

D Optimization Algorithm

The main goal of the optimization algorithm is to determine the locations of the segments (their end points) that best approximate the tracking data. In the Bayesian formalism this is equivalent to finding the locations of the end point that maximize the posterior probability of the fit.

In the most general case, this is a multidimensional optimization problem. Each end point has two coordinates that have to be determined. So, the problem of fitting the tracking data by n connected line segments has $2(n+1)$ variables. This means that to find the optimal fit we have to find a maximum on a $2(n+1)$ dimensional landscape. For example, fitting a tracking series by ten segments requires finding a maximum on a 22 dimensional landscape.

For the problem we are solving it is not practical to use general purpose algorithms for multidimensional optimization because of their complexity and high computational requirements for large values of n. Instead we have developed a less computationally intensive special purpose optimization algorithm. The main idea behind this algorithm is to optimize the locations of the end points one by one instead of simultaneously optimizing the locations of all the end points. The algorithm operates by choosing one of the end points at random and optimizing its location while keeping locations of all the other end points fixed. Then one of the remaining end points is chosen, again at random, and its location is optimized. This process continues until locations of all the end points are optimized (individually). At this point the new posterior probability is computed and compared

to the value of the posterior probability before the optimization pass. If the relative increase in the posterior probability is less than a threshold value, the optimization procedure stops. Otherwise another optimization pass is performed, i.e., locations of all the end points are again individually optimized (in random order). This is repeated until the procedure converges when the relative increase of the posterior probability produced by the last optimization pass becomes smaller than the threshold.

A cartoon illustrating the optimization procedure that was just described is shown in Fig. 10. Here a set of noisy data is being fit by five connected line segments. The steps taken by the optimization procedure are numbered on the left of the cartoon. At every step the location of only one end point is optimized (marked by arrows).

Since we are working in a two dimensional space (length versus time, or x versus y), every end point has two coordinates. So, finding the optimal location of an end point is a maximization problem in two dimensions. It is solved using the Polak-Ribiere variant of the Fletcher-Reeves procedure described in (1).

E Fitting Procedure in Detail

Going from many short segments to a few long segments

The fitting procedure starts by fitting the data by many very short segments, and then gradually reducing the number of segments by repeatedly merging pairs of adjacent segments and optimizing the locations of the remaining segments. This process continues until the desired number of segments is reached or, alternatively, until some other criterion such as reaching a predefined value of the posterior probability is met.

At every iteration we must chose which pair of adjacent segments will be merged. In most cases, for a given number of segments, there is only one optimal configuration. The ideal optimization algorithm would be able to find this configuration starting from any random configuration. In practice any multidimensional optimization can get stuck in a local minimum or, in the case of our limited optimization routine, in a configuration that is pseudo-optimal. Such a configuration is not optimal in the strict sense, but cannot be improved by using the limited optimization algorithm. The number of computations required to find the optimal configuration is also affected by the initial configuration. So, the best approach is to merge a pair of segments that is likely to result in a configuration that is close to optimal (for the system with the number of segments reduced by one). Using this configuration as the starting point for the optimization algorithm will help it to quickly converge to the optimal configuration.

A quantitative measure for finding a pair of segments to merge

To find a pair of adjacent segments that is a good candidate for merging we look at how much the segmented line would be distorted if the shorter of the two segments was eliminated and the

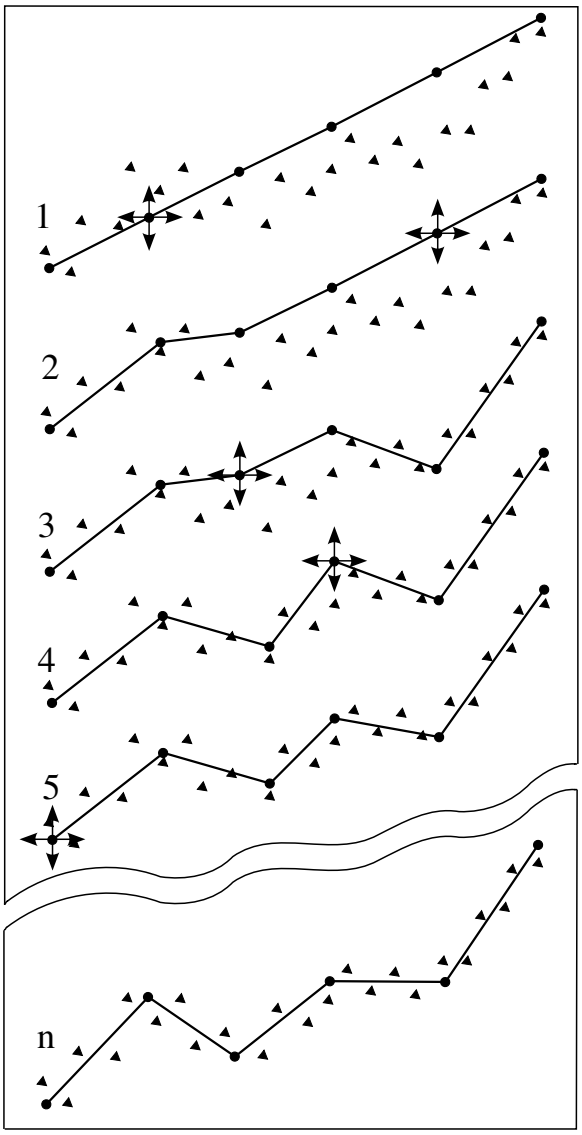

Figure 10: Illustration of the optimization procedure. A ficticious set of noisy tracking data depicted by the triangles is being fit by five connected line segments. Steps taken by the optimization procedure are shown (marked by numbers). At each step the location of one end point (marked by arrows) is optimized while the rest of the end points are fixed.

duration of the longer segment was increased to equal the sum of the durations of the segments in the pair. (Think of a triangle where two of the sides are the pair of segments and the third side is the sum of the durations of the segments in the pair.) A quantitative measure of this distortion can be defined as the product of the absolute value of the difference of velocities of the segments and the duration of the shorter of the two segments. The pair that should be merged has the lowest value of this measure. Using this procedure we can quickly identify the pair of segments that should be merged. Once the pair of segments has been identified, they are merged by eliminating the common vertex. In other words, if the two segments are the sides of a triangle, the third side of the triangle would be the result of merging the segments. To achieve maximum accuracy we merge several likely candidate pairs and choose the one that produces the highest posterior probability.

Next, we give a more detailed description of the fitting procedure that uses the approaches outlined above. Assume that we have a tracking series that consists of N data points. The task is to fit it by n connected line segments. Here we assume that n is a known value less than N. In the next subsection we shall discuss approaches for determining the value of n.

Initializing the fitting procedure

During the first iteration of the fitting procedure the tracking data is fitted by k connected line segments. The value of k is selected to be much larger than n. In the extreme case k is equal to $N-1$, so that there is one segment per data point.

The following recursive procedure is applied to define the initial positions of the line segments. First, just one segment is created with the end points coinciding with the first and the last data point. Then the segment is split in two parts by identifying the data point that has the largest perpendicular distance away from the segment and using it as a new vertex. The procedure continues in the same fashion until k segments are created. This procedure is designed to provide the initial vertices that are close to optimal.

The final step of the first iteration is to run the optimization algorithm describe earlier to get the optimal fitting of the tracking series by k connected line segments.

Reducing the number of segments

In the following iterations the number of segments is gradually reduced until it becomes equal to n. Each iteration consists of selecting a pair of segments that are the best candidates for merging using the approach described earlier, merging the two segments by eliminating the vertex connecting the segments, and optimizing the locations of the remaining vertices. So, if the starting point of an iteration is the tracking data fitted (optimally) by some number of segments i ($k \geq i > n$), then the result of the iteration is the tracking data fitted by $i-1$ segments. The procedure stops when $i-1=n$.

F Model Selection Without Using a Calibration Curve

As stated before additional information about the nature of the noise present in the tracking data is required to establish the number of constant velocity states (segments) present in a track. Our method of choice for obtaining such information is to use tracking data with no underlying motion to construct a calibration curve that is used later to determine the number of distinct states present in the tracks with unknown underlying motion of a motor complex. In this case the prior probability simply prevents segments with negative duration. However, in some cases it is impossible to obtain suitable calibration data. In this appendix we describe how the prior probability can be used to handle such situations.

If calibration data is not available, one has to make some assumptions about how the thermal fluctuations of the position of the cargo with respect to the position of the motor affect the tracking data. Due to the presence of the fluctuations in the tracking data a single long state of constant velocity motion of the motor complex can be interpreted as several shorter segments, so it is safe to assume that compared to longer (in duration) segments, short segments are more likely to be the result of the thermal fluctuations. To compensate for this effect, a special term can be introduced into the prior probability distribution. Its value is a product over all the segments. The ith term in this products depends on the probability of the duration of the ith segment and the probability that the positions of the points in the ith segment are the result of the underlying motion and not the result of thermal fluctuations.

It is impossible to obtain an exact expression for such a term since it would depend on the unknown properties of the underlying motion and the thermal fluctuations. So one has to seek an approximate expression by making assumptions on how the parsing procedure is affected by the fluctuations. An example of this approach is presented below.

Let us make the following two assumptions. First, assume that there is a constant probability q that the location of a particular data point is affected by the fluctuations so much that it causes the velocity of a segment detected by the program to be significantly different from the velocity of the underlying motion. Second, assume that if several consecutive data points are affected by a fluctuation, they are interpreted as one segment. In the same fashion, several consecutive points that are not affected by a fluctuation are also interpreted as a single segment. However, every pair of data points, one of which is affected by a fluctuation and the other is not, leads to creation of a new segment. For example, a sequence of nine data points, of which the first three are not affected by a fluctuation, the next three are affected, and the last three are again not affected, will be interpreted as three separate segments.

Now, suppose that a segment of duration d that corresponds to n data points was detected. What is the probability that a segment of this duration was caused by a fluctuation? Since in a segment that is caused by fluctuations, all the data points have to be affected by fluctuations, this probability is equal to q^n. It is higher for shorter segments which reflects our assumption that short

segments are more likely to be caused by fluctuations. Earlier we have mentioned that it is possible to compensate for this effect by multiplying the prior probability distribution by a value equal to the probability that the segment in question was not caused by fluctuations. In the present case this probability is equal to $1-q^n$.

This expression can be easily generalized to the case of a set of several segments. The prior probability distribution of a set of segments is given by a product of the prior distributions of individual segments, so for a set of k segments, the prior probability distribution would have the following form:

$$P = \prod_{i=1}^{k} p(d_i)(1-q^{n_i}), \tag{16}$$

Here $p(d_i)$ is equal to zero if the duration (d_i) of the ith segment is negative or less than a specified limit, and is equal to one otherwise. n_i is the number of data points in the ith segment.

We should emphasize that the value of the parameter q cannot be determined from the data and has to be obtained by using some external information such as tracking data with known underlying motion of the molecular motor complex.

In the form presented above the prior probability distribution weights longer segments while the likelihood function is larger for shorter segments since shorter segments fit the data at least as well as longer segments. This means that the posterior probability distribution that is given by Bayes' theorem Eq. 1 is maximized by choosing a model with a finite number of segments. This model is the final answer to the problem that we are solving.

G Derivation of the numerical measure of parsing accuracy

In testing the parsing program, the accuracy of the parsing is given by the probability that a randomly selected sequence of three consecutive segments corresponded to a valid set of states of the underlying motion (state 1/state 2/state 1 or state 2/state 1/state 2). The procedure that was used to compute this probability is presented in this appendix.

First, we find the probability that a given segment corresponds to the first state (p_1) or to the second state (p_2). This is done by assuming that each state is associated with a Gaussian distribution of segment velocities that was constructed earlier. This leads to the following expressions for p_1 and p_2:

$$p_1 \propto P(v, v_1, \sigma_1), \tag{17}$$

$$p_2 \propto P(v, v_2, \sigma_2). \tag{18}$$

Here $P(v, v_m, \sigma)$ is a probability density of a Gaussian distribution with mean v_m and standard deviation σ, and v is the segment velocity. Combining this expression with the normalization

condition

$$p_1 + p_2 = 1 \tag{19}$$

we arrive at

$$p_1 = \frac{P(v, v_1, \sigma_1)}{P(v, v_1, \sigma_1) + P(v, v_2, \sigma_2)}, \tag{20}$$

$$p_2 = \frac{P(v, v_2, \sigma_2)}{P(v, v_1, \sigma_1) + P(v, v_2, \sigma_2)}. \tag{21}$$

The probability that a set of three consecutive segments represents a valid sequence of underlying states (state 1/state 2/state 1 or state 2/state 1/state 2) is then defined as

$$p^i = p_1^i p_2^{(i+1)} p_1^{(i+2)} + p_2^i p_1^{(i+1)} p_2^{(i+2)}. \tag{22}$$

Here i is the index of the first segment in the sequence. The probability that any set of three consecutive segments represents a valid sequence is given by the average of p^i:

$$p = \frac{1}{N} \sum_{i=1}^{N} p^i = \frac{1}{N} \sum_{i=1}^{N} \left[p_1^i p_2^{(i+1)} p_1^{(i+2)} + p_2^i p_1^{(i+1)} p_2^{(i+2)} \right], \tag{23}$$

where N is the total number of three consecutive segment combinations found by the parsing program. We use the value of p as a numerical measure of the accuracy of parsing program.

References

1. Willian H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C. The art of scientific programming*. Cambridge University Press, 2002.