THE JOURNAL OF PHYSICAL CHEMISTRY B

Article

pubs.acs.org/JPCB

A Prüfer-Sequence Based Algorithm for Calculating the Size of Ideal Randomly Branched Polymers

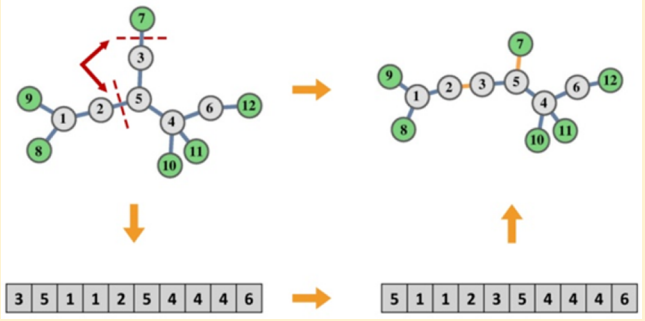
Surendra W. Singaram,[†,‡] Ajaykumar Gopal,[†,‡] and Avinoam Ben-Shaul*[,†]

[†]Institute of Chemistry and the Fritz Haber Research Center, Givat Ram Safra Campus, The Hebrew University, Jerusalem 91904, Israel

[‡]Department of Chemistry, University of California, Los Angeles, California 90095, United States

Supporting Information

ABSTRACT: Branched polymers can be represented as tree graphs. A one-to-one correspondence exists between a tree graph comprised of N labeled vertices and a sequence of $N - 2$ integers, known as the Prüfer sequence. Permutations of this sequence yield sequences corresponding to tree graphs with the same vertex-degree distribution but (generally) different branching patterns. Repeatedly shuffling the Prüfer sequence we have generated large ensembles of random tree graphs, all with the same degree distributions. We also present and apply an efficient algorithm to determine graph distances directly from their Prüfer sequences. From the (Prüfer sequence derived) graph distances, 3D size metrics, e.g., the polymer's radius of gyration, R_g, and average end-to-end distance, were then calculated using several different theoretical approaches. Applying our method to ideal randomly branched polymers of different vertex-degree distributions, all their 3D size measures are found to obey the usual $N^{1/4}$ scaling law. Among the branched polymers analyzed are RNA molecules comprised of equal proportions of the four—randomly distributed—nucleotides. Prior to Prüfer shuffling, the vertices of their representative tree graphs, these "random-sequence" RNAs exhibit an $R_g \sim N^{1/3}$ scaling.



1. INTRODUCTION

Branched polymers are commonly used in chemical industry and are of widespread use in everyday life.[1] There are also various kinds of natural branched biopolymers, such as glycogen, starch, and other polysaccharides.[2] The configurational statistics of branched polymers has been the theme of many physical theories, focusing generally on the scaling relationship $R_g \sim N^{\nu}$ between the radius of gyration of the polymer, R_g, and the number of its constituent monomers, N. For ideal randomly branched polymers, that is, ignoring excluded volume interactions, several different elegant approaches[3−6] revealed that in three dimensions (3D) $\nu = 1/4$, as compared to the less compact ideal linear polymers, for which $\nu = 1/2$. When excluded volume interactions are taken into account, allowing only self-avoiding chain conformations, both linear and branched polymers swell, resulting in $\nu = 3/5$ for linear polymers[5−8] and $\nu = 0.45 \pm 0.06$ for the randomly branched polymers.[9]

Single-stranded (ss) RNA molecules fold into branched polymer structures composed of rigid double stranded duplexes of several base-pairs (bps) connected by flexible single-stranded (ss) loops of nucleotides (nts). For the rather hypothetical but theoretically interesting case of "random-sequence RNAs" which are assumed to be comprised of randomly distributed and equal proportions of the four nts, the duplexes consist of about 5 bps and the loops contain about 10 nts, on average.[10,11] Interestingly, many viral RNAs share similar duplex and loop sizes, yet their branching pattern is qualitatively different.[12−15] The branching pattern of loops and duplexes defines the secondary structure of the ssRNA molecule. The three-dimensional (3D) size of ssRNAs, especially large ones, is conveniently estimated by first mapping their branched secondary structure to tree graphs,[13−16] with the flexible loops regarded as graph vertices and the rigid duplexes as edges (bonds) connecting neighboring vertices. Assuming ideal polymer behavior, the 3D size of the molecule is then obtained by taking the square root of the average contour length between pairs of tree graph ends (corresponding to hairpin pairs in RNA secondary structures). A similar approach yields the $R_g \sim N^{1/4}$ behavior of ideal randomly branched polymers.[6]

Applying the tree graph representation to analyze the secondary and tertiary structures of ssRNA molecules, it was shown that long RNAs comprised of random nucleotide sequences exhibit an $R_g \sim N^{1/3}$ scaling,[12−15] indicating intermediate compactness between the 3D size of ideal linear polymers and randomly branched polymers. It was also found

Special Issue: William M. Gelbart Festschrift

Received: March 3, 2016
Revised: April 18, 2016
Published: April 22, 2016

DOI: 10.1021/acs.jpcb.6b02258
J. Phys. Chem. B 2016, 120, 6231−6237

that viral RNAs (of icosahedral viruses) are consistently more compact than nonviral (e.g., yeast) and random RNAs.[12,13] This conclusion has very recently been confirmed by fluorescence correlation spectroscopy measurements of RNA hydrodynamic radii.[17] Further, an interesting recent theoretical analysis reveals that the compactness of the viral RNA sequences is highly specific; it is lost upon synonymous mutations, namely, mutations that preserve the protein coding sequence but are not found in the virus.[18,19]

Every tree graph comprised of N labeled vertices can be uniquely represented by an ordered sequence of $N - 2$ numbers, known as the Prüfer sequence.[20,21] Conversely, given the Prüfer sequence (P-sequence in short), we can uniquely construct the tree graph from which it was derived. Any permutation of the $N - 2$ elements of the P-sequence produces another sequence and thus (generally) a topologically different tree graph, yet its vertex-degree distribution (or, in short, degree distribution) is identical to that of the original tree graph. The degree distribution is specified by the numbers, n_d ($\sum_d n_d = N$), of vertices of degree d in the graph, i.e., vertices bonded to d neighboring vertices. Equivalently, following normalization, the degree distribution is given by the fractions $\{p_d = n_d/N\}$ of vertices of degree d comprising the tree graph. Combined with Monte Carlo sampling, we have recently used the invariance of $\{n_d\}$ with respect to P-sequence shuffling, in order to compare the R_gs of compact and extended tree graphs with identical degree distributions.[15]

In this paper, we employ Prüfer-sequence permutations—Prüfer-shuffling in short—as a means to describe and analyze the configurational statistics of ideal randomly branched polymers. We shall show that starting from an arbitrarily branched polymer (e.g., a Cayely tree or a tree representing the secondary structure of RNA) repeated shuffles of its P-sequence provide an efficient way to generate large ensembles of randomly branched polymers. More significantly, we shall show that their structural metrics, such as vertex-to-vertex distances, and thus the scaling behavior of 3D-size measures such as R_g can be directly and efficiently derived from their P-sequences.

In the next section (section 2), we introduce our nomenclature for the various types of vertices appearing in our tree graph analyses, briefly outline the definition and derivation of the Prüfer sequence, and describe the shuffling procedure. Then, in section 3, we describe our algorithm for recovering the tree graph from its P-sequence. The derivation of the P-sequence of a given tree graph as well as its recovery from its sequence are both well-known. Here, however, for the sake of eventually deriving size metrics, we employ a specific, simpler, procedure for labeling the tree graph vertices, enabling a simpler (as far as we know new, albeit specific for our purposes) algorithm for recovering the tree graph from its P-sequence. Then, in section 4, we describe our algorithm for deriving secondary structure metrics, e.g., the graph diameter, *directly* from the P-sequence. In section 5, we outline the methods (some well-known and some less known) that we use for deriving 3D properties from the (1D) graph metrics. Numerical results comparing our approach to derive graph metrics directly from the P-sequence with other methods are given in section 6. Concluding remarks are outlined in section 7.

2. PRÜFER SHUFFLING OF TREE GRAPHS

The Prüfer sequence corresponding to a particular, arbitrarily labeled, tree graph is straightforward to construct, as illustrated in Figure 1. The sequence is formed by successive steps; each one involves the removal of the outermost vertex—hereafter the *leaf*—labeled by the smallest number. The vertex to which the plucked leaf was connected—henceforth, the *stump*—is then added as the next element of the P-sequence. This procedure continues until only two (necessarily bonded) vertices are left. One of these vertices is possibly a leaf in the original tree. The other one is necessarily the last element of the P-sequence; it is obviously a *skeletal* vertex (i.e., a $d \geq 2$ vertex) of the original tree. None of the $N - 2$ elements of the P-sequence represents a leaf, because all leaves (except possibly one which may be a member of the last pair) were deleted in the process of constructing the P-sequence.

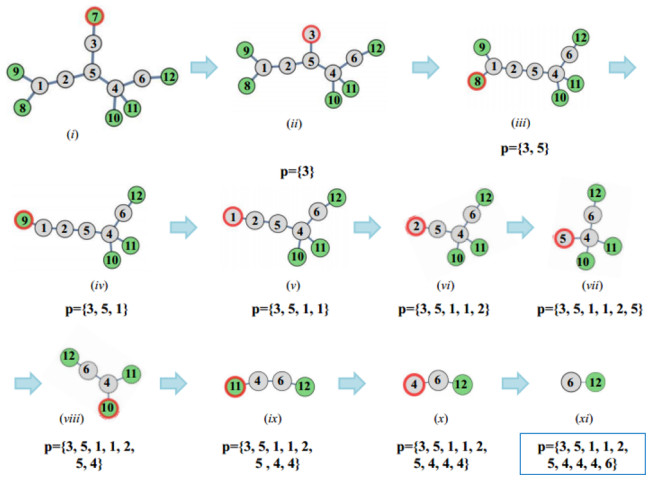


Figure 1. From tree graph to Prüfer sequence. The sequence is generated by successively removing the peripheral vertex ("leaf") bearing the smallest label (circled in red), and adding the number of the vertex to which it was connected as the next element in the P-sequence. Leaves of the original tree are colored green.

While leaves do not appear in the P-sequence, the sequence includes all the skeletal vertices, each of which appearing there $d - 1$ times, because it takes the removal of $d - 1$ vertices before turning a vertex of degree d into a leaf. Thus, since there are $N - 2$ elements in the P-sequence, we have

$$\sum_{d\geq 2} n_d(d - 1) = N - 2 \tag{1}$$

consistent with the Euler's "sum rule" of tree graphs

$$\sum_{d\geq 1} n_d d = 2N - 2 \tag{2}$$

The proof of the last equality is simple: The N vertices of a tree graph are connected by a total of $N - 1$ edges. A vertex of degree d is connected to d edges, each of which is shared by another vertex. The total number of edges is thus $\sum_d n_d d/2$ which, in turn, is equal to $N - 1$, leading directly to eq 2. We also note that the number of leaves (i.e., $d = 1$ vertices) is $n_1 = (2N - 2) - \sum_{d\geq 2} n_d d$. Two limiting cases of interest are the linear and (the maximally branched) star-like graphs, representing the spatially maximally extended and most compact structures, respectively. In both cases, there are only two kinds of vertices. In the linear tree graph, $n_1 = 2$ and $n_2 = N - 2$, whereas, for the star-like graph, $n_1 = N - 1$ and $n_{N-1} = 1$.

Any permutation of the $N - 2$ elements of the Prüfer sequence of a particular labeled tree graph yields a sequence describing another labeled tree graph, with identical degree

DOI: 10.1021/acs.jpcb.6b02258
J. Phys. Chem. B 2016, 120, 6231−6237

distribution, $\{n_d\}$. The branching pattern of the new tree is generally different from that of the original one, as illustrated in Figure 2. Notice that the leaves are not involved in any

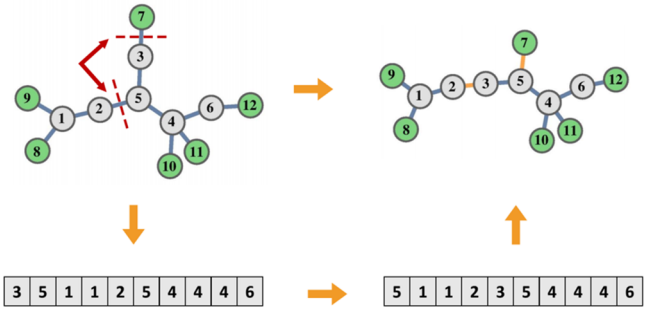


Figure 2. Prüfer shuffle: Permutation of the P-sequence generates a sequence describing a tree graph with the same vertex-degree distribution. The shuffle corresponds to a swap of two branches of the tree graph.

permutation, because they do not appear in the P-sequence. Repeated random shuffles generate an ensemble of sequences corresponding to an ensemble of tree graphs, which obey the configurational statistics of ideal randomly branched polymers, all with the same distribution of vertex-degrees.

3. RECOVERING THE TREE GRAPH FROM ITS PRÜFER SEQUENCE

There are several known ways to recover a tree graph from its Prüfer sequence.[20] Below we outline our own version for achieving this goal. One variant of our approach, which greatly facilitates the generation of tree graph ensembles and calculating graph metrics, is to assign leaves the largest numbers of vertex labels. That is, in a tree of N vertices containing L leaves, the leaves are labeled $N - L + 1$, ..., N (as in Figures 1 and 2). Other than that, the labeling of the L leaves, as well as of the $N - L$ skeletal vertices, is arbitrary. Assigning the highest numerals to the leaves has no effect on the statistics of the tree graph ensembles, because only skeletal vertices are involved in the Prüfer shuffling procedure. One necessary consequence of this assignment convention is that the leaf bearing the highest numeral is never removed, and thus must be a member of the pair of vertices left unplucked upon completing the P-sequence. The other one is necessarily a skeletal vertex.

Our procedure for recovering a tree graph from its P-sequence involves successive inverse moves—from the last element to the first element of the sequence. We shall conveniently refer to the inverse (right-to-left) Prüfer sequence as the P′-sequence. The simplest way to describe this algorithm is by way of example. In Figure 3, we describe, step by step, the recovery of the 12-vertex tree graph shown in Figure 1 from its 10-element P-sequence. Labeling the growing tree from i to xi, we proceed as follows.

(i) The first element in the P′-sequence in Figure 3 is vertex #6 (V6 in short). This is the last skeletal vertex exposed by removing an adjacent vertex, and is one of the last two vertices left upon the completion of the steps leading to the P-sequence corresponding to the tree graph of Figure 1 ({3,5,1,1,2,5,4,4,4,6}). By our convention of assigning the L largest numbers to the leaves of the original tree, the other member of this last pair is the leaf labeled by the largest number, i.e., V12. Our "seed" of the growing tree—tree graph i in Figure 3—thus consists of

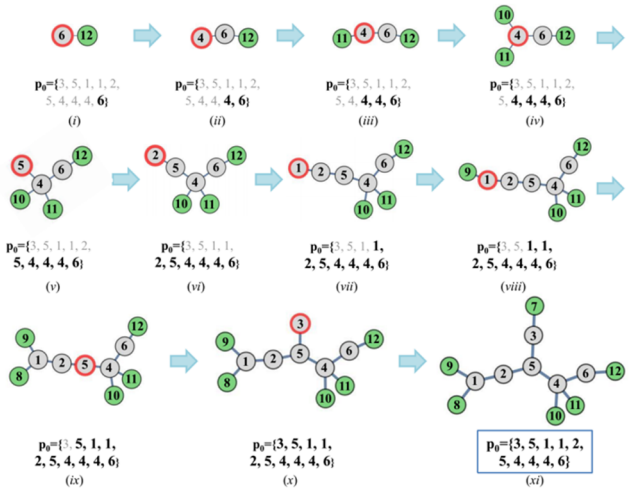


Figure 3. From Prüfer sequence to tree graph. Leaves of the original tree are colored green, and skeletal vertices are in gray. The red ring highlights the reactive vertices of the growing tree. Whenever a skeletal vertex is added to the tree, its label is boldfaced in the sequence. See the text for the detailed description.

the leaf V12 (colored green) and the skeletal (stump) vertex V6 (colored gray). In parallel, we boldface V6 in the P′-sequence, marking its addition as the first element of the P′-sequence. We highlight V6 by a red ring, indicating that it is now *reactive*, since it requires at least one more edge to be *saturated*.

(ii) In position 2 of the P′-sequence, we find V4, indicating that its removal led to exposing V6. Thus, in the growing tree graph, we now connect V4 to V6 and highlight it red as the next reactive vertex. In parallel, we boldface V4 in the P′-sequence.

(iii) Next, in position 3, we find V4 again. This means that V4 in position 2 necessarily arrived there following the removal of a leaf, which must be the leaf with the highest index available, namely, V11. This also means that V4 is a stump. We boldface it in the sequence and move to position 4.

(iv) In position 4, we find V4 again, implying that V4 in position 3 is the result of plucking yet another leaf, i.e., V10. Thus, V4 is a 2-fold stump. We boldface the third V4 as well.

(v) In position 5 of the P′-sequence stands V5, implying that its removal is responsible for the appearance of V4 in position 4. We thus add V5 to the tree graph, connecting it to V4, highlighting it by a red ring as the next reactive vertex. In the P′-sequence, we boldface V5, and procced to the next element in the sequence.

(vi) In position 6, we find V2, concluding that its removal leads to V5 in position 5. Connecting V5 to V2 in the graph, boldfacing the latter in the sequence and highlighting it in the graph, we move to position 7, finding there V1.

(vii) In the graph, we now connect V1 to V2 and assign V1 the reactive red ring label. Boldfacing V1 in position 7 of the P′-sequence, we move to the next element.

(viii) In position 8, we find V1 again. As argued earlier with respect to V4, this means the V1 in position 7 appeared there by plucking a leaf, namely, V9. We note that V1 is a stump. In position 8 of the P′-sequence, we boldface V1.

DOI: 10.1021/acs.jpcb.6b02258
J. Phys. Chem. B 2016, 120, 6231−6237

(ix) The next element of the sequence in position 9 is V5. Since V5 is already included in the growing tree, the presence of V1 in position 8 cannot be due to the deletion of V5. V1 in position 8 is thus due to leaf deletion, i.e., V8, and V1 is thus a 2-fold stump. Importantly, the recurring appearance of vertex V5 indicates the emergence of a new branch emanating from V5. The presence of V3 in position 10 indicates that this branch begins with the stem connecting V5 and V3. We boldface V5 in position 9, and move to the next stage.

(x) In the last element of the P′-sequence stands V3, whose removal led to V5 in position 9. In the tree graph, we connect V3 to V5, and boldface it in the sequence. Being the last element in the P′-sequence (first in the P-sequence), V3 is a stump. It was exposed by the deletion of a leaf—the last leaf yet unassigned, namely, V7. We add it to the tree graph, thus completing its recovery from the P-sequence.

4. FROM P-SEQUENCE TO GRAPH METRICS

In this section we show how to derive tree graph metrics from P-sequences, e.g., the leaf-to-leaf distance (LLD)—expressing the contour distance (i.e., number of edges) between a pair of leaves, or the graph diameter (GD)—representing the maximal leaf-to-leaf distance. On the basis of these "secondary structure" properties, one can derive average 3D properties such as the radius of gyration of a branched polymer, or its average LLD, as discussed in more detail in the next section. The common numerical route to derive properties of large ensembles of tertiary structures involves computer simulations. In this section, we show that an alternative, in some ways more efficient, route to achieve this goal is by deriving secondary structure properties directly from P-sequence ensembles. Below, using again the tree graph of Figure 3 as an example, we show how LLDs can be derived from its P-sequence: {3,5,1,1,2,5,4,4,4,6}.

The number of leaf-to-leaf paths in a tree graph containing L leaves is $k = L(L-1)/2$. The contour distance between any pair of leaves is LLD = SSD + 2, with SSD denoting the distance (number of edges) between their respective stumps. (Note that this relation is also valid in the cases of two leaves emanating from the same stump, in which case, by definition, SSD = 0, and hence LLD = 2.) For the tree graph in Figure 1, $L = 6$ and the number of leaf-to-leaf paths is thus $k = 15$, several of which are shown in Figure 4. In the previous section, analyzing the P′-sequence of this tree graph, we found that its six leaves are connected to four stumps: V6, V3, and the two 2-fold stumps V4 and V1. Boldfacing the stumps' six positions in the Prüfer sequence, {**3**,5,**1**,**1**,2,5,4,**4**,**4**,**6**}, we now proceed to unfold the 15 leaf-to-leaf paths embodied in this sequence.

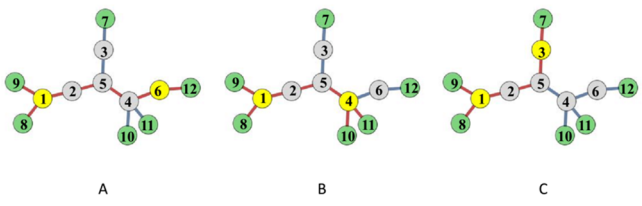


Figure 4. Examples of leaf-to-leaf paths (colored red). Leaves are colored green, and stumps are colored yellow. See text for details.

To identify the leaf-to-leaf paths, we refer again to the P′-sequence. For convenience, we start with paths originating from the rightmost stump, and then progress leftward from one stump to the next.

(i) Paths originating from V6:
  (a) As already known from the analysis in the previous section, the stump V6 is connected to V4, which is also a stump, actually a 2-fold stump. Thus, two leaf-to-leaf paths symbolized as $(6 \rightarrow 4) \times 2$ lead from V6 to V4. Their length is LLD = 3.
  (b) V4 is further connected to V5, which is then connected to V2 which is finally connected to the 2-fold stump V1. We thus identify two paths of length LLD = 6, namely, $(6 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 1) \times 2$ (Figure 4A).
  (c) Further down the P′-sequence, we note that (in addition to its bonds with V4 and V2) V5 is also connected to the V3 stump, implying another path originating in V6, namely, $(6 \rightarrow 4 \rightarrow 5 \rightarrow 3)$ whose length is LLD = 5 (Figure 4B).

(ii) Paths originating from V4:
  (a) The (4th degree) V4 is a 2-fold stump, implying one short (LLD = 2) path between its two daughter leaves. This path will be symbolized as $(4\rightleftarrows)$.
  (b) The 2-fold stump V4 is also connected to V5 and then to V3, implying two paths $(4 \rightarrow 5 \rightarrow 3) \times 2$ of length LLD = 4.
  (c) The P′-sequence also reveals four "degenerate" paths connecting the two 2-fold stumps V4 and V1, namely, $(4 \rightarrow 5 \rightarrow 2 \rightarrow 1) \times 2 \times 2$ with length LLD = 5 (Figure 4C).

(iii) Paths originating from V1:
  (a) As a 2-fold stump, V1 involves a short (LLD = 2) path $(1\rightleftarrows) \times 1$.
  (b) Somewhat harder to identify, because they involve backward moves along the P′-sequence, are the two LLD = 5 paths $(1 \rightarrow 2 \rightarrow 5 \rightarrow 3) \times 3$.

The total number of leaf-to-leaf paths corresponding to the tree graph considered here is, indeed, $k = 15$. We also find that its average leaf-to-leaf distance is $\overline{\mathrm{LLD}} = 4.33$ and its graph diameter is GD = 6. In section 5, we employ the analysis described here to calculate the ensemble averages $\langle \overline{\mathrm{LLD}} \rangle$ and $\langle \mathrm{GD} \rangle$, and discuss their dependence on polymer size, N.

5. FROM GRAPH METRICS TO 3D SIZE

The mean square radius of gyration, $\overline{R_g^2}$, representing the average of R_g^2 over all possible conformations of an ideal polymer, whether branched or linear, can be calculated using Kramers theorem[5]

$$\overline{R_g^2} = \frac{b^2}{N^2} \sum_{k=1}^{N} N_1(k)[N - N_1(k)] \tag{3}$$

where N is the number of monomers comprising the polymer and b is the monomer−monomer bond length. The summation extends over all possible divisions ("bond breaking") of the polymer into two parts, containing $N_1(k)$ and $N - N_1(k)$ monomers, respectively. Hereafter, for notational brevity, we shall replace $\sqrt{\overline{R_g^2}}$ by R_g. For the special simple case of an ideal linear polymer, eq 3 yields the well-known relationship $R_g \sim N^{1/2}$, identical to the power law dependence of the average

DOI: 10.1021/acs.jpcb.6b02258
J. Phys. Chem. B 2016, 120, 6231−6237

end-to-end distance $\bar{R}$; $\bar{R} = \sqrt{6}R_g$ for a freely jointed linear chain.[5] For ideal randomly branched polymers, eq 3 yields the familiar theoretical scaling law, $R_g \sim N^{1/4}$. The average end-to-end distance of an ideal randomly branched polymer is equivalent to our average leaf-to-leaf distance, $R_{LL} \equiv \sqrt{\langle \overline{LLD} \rangle}$ and should thus also scale as[6] $N^{1/4}$. This behavior will be confirmed in the next section, using our Prüfer algorithm. A related quantity showing the same power law behavior is $R_{GD} \equiv \sqrt{\langle GD \rangle}$, the square root of the graph diameter.

Before proceeding to the numerical results, we mention that the metrics $\langle \overline{LLD} \rangle$ and $\langle GD \rangle$ are analogous to two measures that have formerly been used to characterize the sizes of ssRNA molecules. As noted in section 1, RNA secondary structures can be mapped onto tree graphs with the vertices representing the flexible single stranded loops of nucleotides and edges representing base pair duplexes. Hairpins along with the external loop are the leaves of the corresponding tree graph. One metric, called "the maximum ladder distance" (MLD), specifying the maximal distance between hairpin loops,[12,22] is proportional to the corresponding RNA graph diameter, GD. Another metric, "the average ladder distance" (ALD),[12] i.e., the average contour distance between loops, is proportional to our leaf-to-leaf distance, $\overline{LLD}$. Interestingly, for random RNA sequences, it was found[12−14] that the ensemble averages of the maximal and average ladder distances, $\langle \overline{MLD} \rangle$ and $\langle \overline{LLD} \rangle$, scale as $N^{2/3}$. In other words, they appear to be less compact than ideal randomly branched polymers with the same distribution of vertex degrees. On the other hand, as noted in section 1, viral RNAs (of icosahedral viruses) are consistently more compact than random sequence RNAs of the same nucleotide composition.[12−14,18,19]

6. RESULTS

Using our Prüfer-sequence algorithm for calculating vertex-to-vertex distances, and the theoretical tools described in the previous section, we have numerically calculated several 3D size measures of ideal randomly branched polymers (IRBP), and compared them to results obtained by more familiar approaches. Numerical results corresponding to two families of IRBPs with different degree distributions, $\{p_d = n_d/N\}$, are shown in Figure 5. Two additional IRBP families revealing the same qualitative behavior are discussed in the Supporting Information.

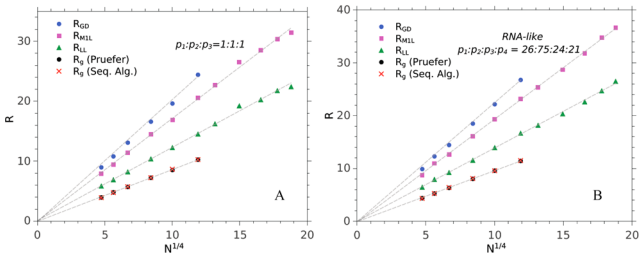


Figure 5. 3D size measures of randomly branched tree graphs as a function of the (fourth root of the) number of vertices N. Parts A and B correspond to different vertex-degree distributions, as indicated in the respective figures. The computation algorithms leading to the five data sets are detailed in the text.

The results in Figure 5 are ensemble averages of size measures for IRBPs comprised of up to N = 120,000 vertices. For every value of N, we have analyzed 100 tree graphs.

Figure 5A shows the results obtained for branched polymers containing (on average) equal proportions of 2-fold (d = 2) and 3-fold (d = 3) vertices, i.e., p_2:p_3 = 1:1. The fraction of leaves in this tree graph family, p_1, follows from the normalization condition $\sum_{d\geq 1} p_d = 1$ and eq 2, which for large N lead to $p_1 = p_2 = p_3 = 1/3$.

Figure 5B describes the results for random "RNA-like" polymers. The vertex-degree distribution corresponding to this family of IRBPs derives from earlier calculations of RNA secondary structure. Specifically, Boltzmann weighted ensembles of 200 secondary structures corresponding to 200 different randomly generated 7000-nt-long random sequence RNAs with uniform nt composition (i.e., a total of 4×10^4 conformations) were generated using the RNAsubopt program from the Vienna package.[23] Analyzing their branching pattern, it was found that fifth or higher order vertices (i.e., nt-loops) are extremely rare.[14] Lumping their fraction into p_4, the degree distribution corresponding to this family (in the limit of large N) is given by p_1 = 26/126, p_2 = 75/126, p_3 = 24/126, p_4 = 1/126. This distribution is not very different from that of viral RNAs, yet the viral-like tree graphs are more compact than those corresponding to the random RNAs.[12,13,15]

The two IRBP families discussed in the Supporting Information are the following: (i) Cayley trees of 3-fold vertices, i.e., trees consisting of vertices of degree 3 or 1 (i.e., leaves), with $p_1 = p_3 = 1/2$. (ii) Similar to the IRBPs in Figure 5A but with a larger fraction of the 2-fold vertices: $p_1 = 1/6$, $p_2 = 2/3$, $p_3 = 1/6$.

The data points of the 3D size metrics shown in Figure 5 represent the statistical averages corresponding to 100 randomly generated tree graphs for each value of N. Their meaning and calculation procedures are as follows:

- R_{GD}: As defined in section 5, $R_{GD} \equiv \sqrt{\langle GD \rangle}$ is the ensemble average of the square root of the graph diameter. In this calculation, starting with an arbitrary tree graph with the given $\{p_d\}$, ensembles of IRBP tree graph were generated using Prüfer shuffle, and their GD determined by a Mathematica[24] built-in function GraphDiameter. This function utilizes well-known algorithms[25−27] to compute graph distances from the adjacency matrix[28] of the tree graph (see below).
- R_g(Prüfer): This metric is the ensemble averaged radius of gyration, calculated using Kramers formula, eq 3. Here too, IRBP tree graphs were generated using Prüfer shuffling, but vertex-to-vertex distances were calculated using a custom program utilizing the adjacency matrix.
- R_g(Seq. Alg.): These calculations of R_g are based on an efficient method—known as the sequential algorithm—for the generation of random trees with a given degree distribution[29] (see the Supporting Information for more details). Once generated, their R_g's were evaluated on the basis of Kramers theorem using our custom program mentioned above.
- R_{LL}: $R_{LL} \equiv \sqrt{\langle \overline{LLD} \rangle}$, where $\langle \overline{LLD} \rangle$ is the ensemble average of $\overline{LLD}$, the mean leaf-to-leaf distance. In calculating R_{LL}, we have used Prüfer shuffing to generate RBP ensembles, and LLDs were evaluated following the analysis of the P′-sequences described in section 3. One simplification employed in this calculation is that in evaluating $\overline{LLD}$ for a given tree graph we have not included all leaf-to-leaf paths but rather only those originating from the first (rightmost) element of the P′-sequence (like those shown in Figure 4A). In other

DOI: 10.1021/acs.jpcb.6b02258
J. Phys. Chem. B 2016, 120, 6231−6237

words, we have sampled only a fraction of all the leaf-to-leaf paths. Note, however, that in the ensemble of randomly generated tree graphs the identity of the vertex appearing in position 1 of the P′-sequence is also arbitrary, implying that the R_{LL}'s shown in Figure 5 are faithful statistical averages of LLDs. The simplification just mentioned facilitates the calculation, as explained in more detail in the Supporting Information.

- R_{M1L}: Similar to R_{GD}, this quantity measures the ensemble average of the maximal 3D end-to-end distance of a branched polymer. We calculated it subject to the same computational simplification employed in our calculation of R_{LL}. Explicitly, from the paths used to calculate R_{LL}, we picked the longest leaf-to-leaf path (originating from the P′-sequence's first element), obtaining the M1LD (see, e.g., Figure 4A). Using ⟨M1LD⟩ to denote the ensemble average of M1LD, we then define $R_{M1L} = \sqrt{\langle M1LD \rangle}$. Clearly, M1LD ≤ GD, and hence $R_{M1L} \leq R_{GD}$, as is apparent from Figure 5. In the Supporting Information, we show that the ⟨GD⟩ can be approximated by the average M1LD corresponding to the top decile of M1LD.

All five measures of the 3D size of the IRBPs in Figure 5 show the expected scaling relation $R \sim N^{1/4}$. The computational procedures yielding the first three measures—R_{GD}, R_g(Prüfer), and R_g(Seq. Alg.)—utilize the adjacency matrix (AM) to represent the branching pattern of the tree graph. For a tree with N vertices, the AM is a sparse $N \times N$ matrix whose ijth element is 1 if a bond connects vertices i and j and 0 otherwise.[28] In terms of CPU time, these calculations are considerably faster than our calculations of R_{LL} and R_{M1L} based on the determination of LLDs from the P′-sequence; see the Supporting Information for details. Nonetheless, the AM based calculations require the computer to allocate $\sim N^2$ bytes of random access memory (RAM) to handle a tree of N vertices. On the other hand, extracting graph distances from a Prüfer sequence of N elements requires the allocation of only $\sim N$ bytes. Thus, while Mathematica's GraphDiameter function and the sequential algorithms methods are faster than our custom Prüfer algorithms, the latter is far more efficient for large values of N. Indeed, as reflected by the range of N values in Figure 5, when the trees became large (N > 20,000 vertices), the AM based calculations (i.e., R_{GD} and R_g) required more memory than was available in our system (see the Supporting Information) and the calculations were forced to terminate, obviating the utility of the GraphDiameter or the sequential algorithm. In contrast, our programs were able to comfortably calculate the LLD and ML1LD well beyond this limit.

7. CONCLUDING REMARKS

The results presented in the previous section show that our Prüfer-based algorithm is slower yet memory-wise more efficient than the other algorithms we have used for calculating branched polymer sizes. It is not unlikely that professional programming could possibly upgrade our "homemade" method to make it both faster and even more memory efficient. Nevertheless, presenting yet another algorithm for calculating the size of randomly branched polymers was not the primary goal in this work. We regard the Prüfer shuffle procedure as an elegant way to compare different types of branched polymeric structures, primarily RNA. The present paper represents a step forward toward our more challenging goal of understanding the branching patterns of random and nonrandom RNA sequences, which are qualitatively different from those of ideal random branched polymers.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jpcb.6b02258.

3D size calculations for two additional families of branched polymers: (i) Cayley trees of 3-fold vertices, i.e., trees consisting of vertices of degree 3 or 1 (i.e., leaves), with $p_1 = p_3 = 1/2$; (ii) trees with degree distribution $p_1 = 1/6$, $p_2 = 2/3$, $p_3 = 1/6$. We also show there how the average GD can be estimated from the average of the top decile of ML1LDs. The details of our system, computational time, and memory usage are also given. Finally, we explain how to sample leaf-to-leaf distances directly from the Prüfer sequence, and present a general method for the calculation of vertex-to-vertex distances. (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: avinoambs@mail.huji.ac.il.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This article is dedicated to our mentor, colleague, and friend Bill (William M.) Gelbart, on the occasion of his 70th birthday. We thank Bill for many years of joint work, for teaching us about the physics and biology of DNA, RNA, viruses, and many other related and less related topics, for his inspiring endless scientific enthusiasm and curiosity, and for being a great guy and always a Mentsch.

■ REFERENCES

(1) LaJeunesse, S. Plastic Bags: Plastic Bags Are Not Created Equal Because They Are Meant for Different Purposes. *Chem. Eng. News* **2004**, *82*, 51.

(2) Stryer, L. *Biochemistry*, 4th. ed.; W.H. Freeman and Company: New York, 1995.

(3) Zimm, B. H.; Stockmayer, W. H. The Dimensions of Chain Molecules Containing Branches and Rings. *J. Chem. Phys.* **1949**, *17* (12), 1301−1314.

(4) de Gennes, P.-G. Statistics of Branching and Hairpin Helices for Dat Copolymer. *Biopolymers* **1968**, *6* (5), 715−729.

(5) Rubinstein, M.; Colby, R. H. *Polymer Physics*, 1st ed.; Oxford University Press: Oxford, U.K., 2003.

(6) Grosberg, A. Y.; Khokhlov, A. R. *Statistical Physics of Macromolecules*; American Institute of Physics: New York, 1994.

(7) Flory, P. J. *Statistical Mechanics of Chain Molecules*; Wiley-Interscience: New York, 1969.

(8) deGennes, P.-G. *Scaling Concepts in Polymer Physics*; Cornell University: Ithaca, NY, 1979.

(9) Redner, S. Mean End-to-End Distance of Branched Polymers. *J. Phys. A: Math. Gen.* **1979**, *12* (9), L239−L244.

(10) Fang, L. T.; Yoffe, A. M.; Gelbart, W. M.; Ben-Shaul, A. A Sequential Folding Model Predicts Length-Independent Secondary Structure Properties of Long ssRNA. *J. Phys. Chem. B* **2011**, *115* (12), 3193−3199.

(11) Yoffe, A. M.; Prinsen, P.; Gelbart, W. M.; Ben-Shaul, A. The Ends of a Large RNA Molecule Are Necessarily Close. *Nucleic Acids Res.* **2011**, *39* (1), 292−299.

DOI: 10.1021/acs.jpcb.6b02258
J. Phys. Chem. B 2016, 120, 6231−6237

(12) Yoffe, A. M.; Prinsen, P.; Gopal, A.; Knobler, C. M.; Gelbart, W. M.; Ben-Shaul, A. Predicting the Sizes of Large RNA Molecules. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105* (42), 16153−16158.
(13) Fang, L. T.; Gelbart, W. M.; Ben-Shaul, A. The Size of RNA as an Ideal Branched Polymer. *J. Chem. Phys.* **2011**, *135* (15), 155105.
(14) Gopal, A.; Egecioglu, D. E.; Yoffe, A. M.; Ben-Shaul, A.; Rao, A. L. N.; Knobler, C. M.; Gelbart, W. M. Viral RNAs Are Unusually Compact. *PLoS One* **2014**, *9*, e105875.
(15) Singaram, S. W.; Garmann, R. F.; Knobler, C. M.; Gelbart, W. M.; Ben-Shaul, A. Role of RNA Branchedness in the Competition for Viral Capsid Proteins. *J. Phys. Chem. B* **2015**, *119* (44), 13991−14002.
(16) Gan, H. H.; Fera, D.; Zorn, J.; Shiffeldrim, N.; Tang, M.; Laserson, U.; Kim, N.; Schlick, T. RAG: RNA-As-Graphs Database-Concepts, Analysis, and Features. *Bioinformatics* **2004**, *20* (8), 1285−1291.
(17) Borodavka, A.; Singaram, S. W.; Stockley, P. G.; Gelbart, W. M.; Ben-Shaul, A.; Tuma, R. Sizes of Long RNA Molecules in Dilute Solution Are Determined by the Branching Patterns of Their Secondary Structures (Preprint).
(18) Tubiana, L.; Bozic, A. L.; Micheletti, C.; Podgornik, R. Synonymous Mutations Reduce Genome Compactness in Icosahedral ssRNA Viruses. *Biophys. J.* **2015**, *108*, 194−202.
(19) Ben-Shaul, A.; Gelbart, W. M. Viral ssRNAs Are Indeed Compact. *Biophys. J.* **2015**, *108* (1), 14−16.
(20) Moon, J. W. Counting Labelled Trees. *Canadian Mathematical Monographs*; Canadian Mathematical Congress: London, 1970; Vol. *1*, pp 1−113.
(21) Prüfer, H. Neuer Beweis Eines Satzes über Permutationen. *Arch. Math. Phys.* **1918**, *27*, 742−744.
(22) Bundschuh, R.; Hwa, T. Statistical Mechanics of Secondary Structures Formed by Random RNA Sequences. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* **2002**, *65* (3), 31903.
(23) Lorenz, R.; Bernhart, S. H.; zu Siederdissen, C.; Tafer, H.; Flamm, C.; Stadler, P. F.; Hofacker, I. L. ViennaRNA Package 2.0. *Algorithms Mol. Biol.* **2011**, *6* (1), 26.
(24) *Mathematica*, version 9.0; Wolfram Research, Inc.: Champaign, IL, 2012.
(25) Dijkstra, E. W. A Note on Two Probles in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269−271.
(26) Floyd, R. W. Algorithm 97: Shortest Path. *Commun. ACM* **1962**, *5* (6), 345.
(27) Warshall, S. A Theorem on Boolean Matrices. *J. Assoc. Comput. Mach.* **1962**, *9* (1), 11−12.
(28) Diestel, R. *Graph Theory*; Springer-Verlag: New York, 2000.
(29) Blitzstein, J.; Diaconis, P. A Sequential Importance Sampling Algorithm for Generating Random Graphs with Prescribed Degrees. *Internet Math.* **2011**, *6* (4), 489−522.

DOI: 10.1021/acs.jpcb.6b02258
J. Phys. Chem. B 2016, 120, 6231−6237