Introduction

Branching networks are useful things:

- **Fundamental to material supply and collection**
  - **Supply**: From one source to many sinks in 2- or 3-d.
  - **Collection**: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction
Introduction

Branching networks are useful things:

- **Fundamental to material supply and collection**
- **Supply**: From one source to many sinks in 2- or 3-d.
- **Collection**: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction
Introduction

Branching networks are useful things:

- Fundamental to material **supply and collection**
- **Supply**: From one source to many sinks in 2- or 3-d.
- **Collection**: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Frame 3/38
Introduction

Branching networks are useful things:

- **Fundamental to material supply and collection**
- **Supply**: From one source to many sinks in 2- or 3-d.
- **Collection**: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...
Introduction

Branching networks are useful things:

- Fundamental to material **supply and collection**
- **Supply**: From one source to many sinks in 2- or 3-d.
- **Collection**: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Examples:
Introduction

Branching networks are useful things:

- Fundamental to material supply and collection
- Supply: From one source to many sinks in 2- or 3-d.
- Collection: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...
Introduction

Branching networks are useful things:

- **Fundamental to material supply and collection**
- **Supply**: From one source to many sinks in 2- or 3-d.
- **Collection**: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction

Branching networks are useful things:

▶ Fundamental to material supply and collection
▶ Supply: From one source to many sinks in 2- or 3-d.
▶ Collection: From many sources to one sink in 2- or 3-d.
▶ Typically observe hierarchical, recursive self-similar structure

Examples:

▶ River networks (our focus)
▶ Cardiovascular networks
▶ Plants
▶ Evolutionary trees
▶ Organizations (only in theory...)

Examples:
Introduction

Branching networks are useful things:

- Fundamental to material **supply and collection**
- **Supply**: From one source to many sinks in 2- or 3-d.
- **Collection**: From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)

Introduction
Definitions
Allometry
Laws
Stream Ordering
Horton’s Laws
Tokunaga’s Law
Nutshell
References
Introduction

Branching networks are useful things:

- **Fundamental to material supply and collection**
- **Supply:** From one source to many sinks in 2- or 3-d.
- **Collection:** From many sources to one sink in 2- or 3-d.
- Typically observe hierarchical, recursive self-similar structure

Examples:

- River networks (our focus)
- Cardiovascular networks
- Plants
- Evolutionary trees
- Organizations (only in theory...)


Branching networks are everywhere...

http://hydrosheds.cr.usgs.gov/
Branching networks are everywhere...
Geomorphological networks

Definitions

- **Drainage basin** for a point \( p \) is the complete region of land from which overland flow drains through \( p \).
- Definition most sensible for a point in a stream.
- **Recursive structure:** Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...
Geomorphological networks

Definitions

- **Drainage basin** for a point $p$ is the complete region of land from which overland flow drains through $p$.
- Definition most sensible for a point in a stream.
- **Recursive structure**: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...
Geomorphological networks

Definitions

- **Drainage basin** for a point \( p \) is the complete region of land from which overland flow drains through \( p \).
- Definition most sensible for a point in a stream.
- **Recursive structure**: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...
Geomorphological networks

Definitions

- **Drainage basin** for a point $p$ is the complete region of land from which overland flow drains through $p$.
- Definition most sensible for a point in a stream.
- **Recursive structure**: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...
Geomorphological networks

Definitions

- **Drainage basin** for a point $p$ is the complete region of land from which overland flow drains through $p$.
- Definition most sensible for a point in a stream.
- **Recursive structure**: Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...
Geomorphological networks

Definitions

- **Drainage basin** for a point $p$ is the complete region of land from which overland flow drains through $p$.
- Definition most sensible for a point in a stream.
- **Recursive structure:** Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...
Geomorphological networks

Definitions

- **Drainage basin** for a point $p$ is the complete region of land from which overland flow drains through $p$.
- Definition most sensible for a point in a stream.
- **Recursive structure:** Basins contain basins and so on.
- In principle, a drainage basin is defined at every point on a landscape.
- On flat hillslopes, drainage basins are effectively linear.
- We treat subsurface and surface flow as following the gradient of the surface.
- Okay for large-scale networks...
Basic basin quantities: $a, l, L_\parallel, L_\perp$:

- $a = \text{drainage basin area}$
- $l = \text{length of longest (main) stream (which may be fractal)}$
- $L = L_\parallel = \text{longitudinal length of basin}$
- $L = L_\perp = \text{width of basin}$
Basic basin quantities: \( a, l, L_{\parallel}, L_{\perp} \):

- \( a = \) drainage basin area
- \( l = \) length of longest (main) stream (which may be fractal)
- \( L = L_{\parallel} = \) longitudinal length of basin
- \( L = L_{\perp} = \) width of basin
Basic basin quantities: $a, l, L_\parallel, L_\perp$:

- $a =$ drainage basin area
- $l =$ length of longest (main) stream (which may be fractal)
- $L = L_\parallel = \text{longitudinal length of basin}$
- $L = L_\perp = \text{width of basin}$
Basic basin quantities: $a$, $l$, $L_\parallel$, $L_\perp$:

- $a = \text{drainage basin area}$
- $l = \text{length of longest (main) stream (which may be fractal)}$
- $L_\parallel = L = \text{longitudinal length of basin}$
- $L_\perp = \text{width of basin}$
Basic basin quantities: $a$, $l$, $L_\parallel$, $L_\perp$:

- $a$ = drainage basin area
- $l$ = length of longest (main) stream (which may be fractal)
- $L = L_\parallel = $ longitudinal length of basin
- $L = L_\perp = $ width of basin
**Allometry**

**Isometry:** dimensions scale linearly with each other.
Isometry: dimensions scale linearly with each other.

Allometry: dimensions scale nonlinearly.
Basin allometry

Allometric relationships:

\[ l \propto a^h \]

\[ l \propto L^d \]

Combine above:

\[ a \propto L^{d/h} \equiv L^D \]
Basin allometry

Allometric relationships:

- \( \ell \propto a^h \)
- \( \ell \propto L^d \)
- Combine above:
  \[ a \propto L^{d/h} \equiv L^D \]
Basin allometry

Allometric relationships:

1. \( \ell \propto a^h \)
2. \( \ell \propto L^d \)

Combine above:

\( a \propto L^{d/h} \equiv L^D \)
Basin allometry

Allometric relationships:

\[ \ell \propto a^h \]
\[ \ell \propto L^d \]

Combine above:
\[ a \propto L^{d/h} \equiv L^D \]
‘Laws’

- Hack’s law (1957)\cite{2}:
  \[ l \propto a^h \]
  reportedly \( 0.5 < h < 0.7 \)

- Scaling of main stream length with basin size:
  \[ l \propto L^d \]
  reportedly \( 1.0 < d < 1.1 \)

- Basin allometry:
  \[ L_{||} \propto a^{h/d} \equiv a^{1/D} \]
  \( D < 2 \rightarrow \) basins elongate.
‘Laws’

- Hack’s law (1957)\(^{[2]}\):
  \[ l \propto a^h \]
  reportedly \(0.5 < h < 0.7\)

- Scaling of main stream length with basin size:
  \[ l \propto L_d \]
  reportedly \(1.0 < d < 1.1\)

- Basin allometry:
  \[ L_\parallel \propto a^{h/d} \equiv a^{1/D} \]
  \(D < 2 \rightarrow\) basins elongate.
‘Laws’

- Hack’s law (1957)\textsuperscript{[2]}:
  \[ \ell \propto a^h \]
  reportedly \(0.5 < h < 0.7\)

- Scaling of main stream length with basin size:
  \[ \ell \propto L^d \parallel \]
  reportedly \(1.0 < d < 1.1\)

- Basin allometry:
  \[ L^\parallel \propto a^{h/d} \equiv a^{1/D} \]

\(D < 2 \rightarrow \) basins elongate.
There are a few more ‘laws’: [1]

<table>
<thead>
<tr>
<th>Relation</th>
<th>Name or description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_k = T_1(R_T)^k$</td>
<td>Tokunaga’s law</td>
</tr>
<tr>
<td>$\ell \sim L^d$</td>
<td>self-affinity of single channels</td>
</tr>
<tr>
<td>$n_\omega/n_{\omega+1} = R_n$</td>
<td>Horton’s law of stream numbers</td>
</tr>
<tr>
<td>$\bar{\ell}<em>{\omega+1}/\bar{\ell}</em>\omega = R_\ell$</td>
<td>Horton’s law of main stream lengths</td>
</tr>
<tr>
<td>$\bar{a}<em>{\omega+1}/\bar{a}</em>\omega = R_a$</td>
<td>Horton’s law of basin areas</td>
</tr>
<tr>
<td>$\bar{s}<em>{\omega+1}/\bar{s}</em>\omega = R_s$</td>
<td>Horton’s law of stream segment lengths</td>
</tr>
<tr>
<td>$L_\perp \sim L^H$</td>
<td>scaling of basin widths</td>
</tr>
<tr>
<td>$P(a) \sim a^{-\tau}$</td>
<td>probability of basin areas</td>
</tr>
<tr>
<td>$P(\ell) \sim \ell^{-\gamma}$</td>
<td>probability of stream lengths</td>
</tr>
<tr>
<td>$\ell \sim a^h$</td>
<td>Hack’s law</td>
</tr>
<tr>
<td>$a \sim L^D$</td>
<td>scaling of basin areas</td>
</tr>
<tr>
<td>$\Lambda \sim a^\beta$</td>
<td>Langbein’s law</td>
</tr>
<tr>
<td>$\lambda \sim L^\varphi$</td>
<td>variation of Langbein’s law</td>
</tr>
</tbody>
</table>
Reported parameter values: [1]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Real networks:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_n$</td>
<td>3.0–5.0</td>
</tr>
<tr>
<td>$R_a$</td>
<td>3.0–6.0</td>
</tr>
<tr>
<td>$R_\ell = R_T$</td>
<td>1.5–3.0</td>
</tr>
<tr>
<td>$T_1$</td>
<td>1.0–1.5</td>
</tr>
<tr>
<td>$d$</td>
<td>1.1 ± 0.01</td>
</tr>
<tr>
<td>$D$</td>
<td>1.8 ± 0.1</td>
</tr>
<tr>
<td>$h$</td>
<td>0.50–0.70</td>
</tr>
<tr>
<td>$\tau$</td>
<td>1.43 ± 0.05</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>1.8 ± 0.1</td>
</tr>
<tr>
<td>$H$</td>
<td>0.75–0.80</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.50–0.70</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>1.05 ± 0.05</td>
</tr>
</tbody>
</table>
Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values
Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values
Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values
Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values
Kind of a mess...

Order of business:

1. Find out how these relationships are connected.
2. Determine most fundamental description.
3. Explain origins of these parameter values

For (3): Many attempts: not yet sorted out...
Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) \(^3\)
- Modified by Strahler (1957) \(^6\)
- Term: Horton-Strahler Stream Ordering \(^4\)
- Can be seen as **iterative trimming** of a network.
Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Can be seen as iterative trimming of a network.
Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Can be seen as iterative trimming of a network.
Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Can be seen as iterative trimming of a network.
Stream Ordering:

Method for describing network architecture:

- Introduced by Horton (1945) [3]
- Modified by Strahler (1957) [6]
- Can be seen as iterative trimming of a network.
Stream Ordering:

Some definitions:

- A **channel head** is a point in landscape where flow becomes focused enough to form a stream.
- A **source stream** is defined as the stream that reaches from a channel head to a junction with another stream.
- Roughly analogous to capillary vessels.
- Use symbol $\omega = 1, 2, 3, \ldots$ for stream order.
Stream Ordering:

Some definitions:

- A **channel head** is a point in landscape where flow becomes focused enough to form a stream.
- A **source stream** is defined as the stream that reaches from a channel head to a junction with another stream.
  - Roughly analogous to capillary vessels.
  - Use symbol $\omega = 1, 2, 3, \ldots$ for stream order.
Stream Ordering:

Some definitions:

- A **channel head** is a point in landscape where flow becomes focused enough to form a stream.
- A **source stream** is defined as the stream that reaches from a channel head to a junction with another stream.
- Roughly analogous to capillary vessels.
- Use symbol $\omega = 1, 2, 3, \ldots$ for stream order.
Stream Ordering:

Some definitions:

- **A channel head** is a point in landscape where flow becomes focused enough to form a stream.
- **A source stream** is defined as the stream that reaches from a channel head to a junction with another stream.
- Roughly analogous to capillary vessels.
- Use symbol $\omega = 1, 2, 3, \ldots$ for stream order.
Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order $= \Omega$)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$. 
Stream Ordering:

1. Label all **source streams** as order $\omega = 1$ and remove.
2. Label all **new** source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order $= \Omega$)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$. 
Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order $= \Omega$)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$. 
Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = $\Omega$)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$. 
Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = $\Omega$).
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$. 
Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order = $\Omega$)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$. 
Stream Ordering:

1. Label all **source streams** as order $\omega = 1$ and remove.
2. Label all **new source streams** as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order $= \Omega$)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$. 
Stream Ordering:

1. Label all source streams as order $\omega = 1$ and remove.
2. Label all new source streams as order $\omega = 2$ and remove.
3. Repeat until one stream is left (order $= \Omega$)
4. Basin is said to be of the order of the last stream removed.
5. Example above is a basin of order $\Omega = 3$. 
Stream Ordering—A large example:

Mississippi

longitude
latitude

$\omega = 11$

-105 -100 -95 -90 -85
30 32 34 36 38 40 42 44 46 48

sources: dodds@ncl.ac.uk/rivernetworks/figures/paths_mispi10.ps
21-Mar-2000 peter dodds
Stream Ordering:

Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).
- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:
  $$\omega_3 = \max(\omega_1,\omega_2) + \delta_{\omega_1,\omega_2}$$
  where $\delta$ is the Kronecker delta.
Stream Ordering:

Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.

- Simple rule:
  
  $$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$$

  where $\delta$ is the Kronecker delta.
Stream Ordering:

Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream
  - Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).
- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:
  $$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1,\omega_2}$$
  where $\delta$ is the Kronecker delta.
Stream Ordering:

Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.
- Simple rule:
  \[ \omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2} \]
  where $\delta$ is the Kronecker delta.
Stream Ordering:

Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.

- Simple rule:
  \[ \omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2} \]
  where $\delta$ is the Kronecker delta.
Stream Ordering:

Another way to define ordering:

- As before, label all source streams as order $\omega = 1$.
- Follow all labelled streams downstream.
- Whenever two streams of the same order ($\omega$) meet, the resulting stream has order incremented by 1 ($\omega + 1$).

- If streams of different orders $\omega_1$ and $\omega_2$ meet, then the resultant stream has order equal to the largest of the two.

- Simple rule:

$$\omega_3 = \max(\omega_1, \omega_2) + \delta_{\omega_1, \omega_2}$$

where $\delta$ is the Kronecker delta.
Stream Ordering:

One problem:

- Resolution of data messes with ordering
  - Micro-description changes (e.g., order of a basin may increase)
  - ... but relationships based on ordering appear to be robust to resolution changes.
Stream Ordering:

One problem:

- Resolution of data messes with ordering
- Micro-description changes (e.g., order of a basin may increase)
- ... but relationships based on ordering appear to be robust to resolution changes.
Stream Ordering:

One problem:

- Resolution of data messes with ordering
- Micro-description changes (e.g., order of a basin may increase)
- ... but relationships based on ordering appear to be robust to resolution changes.
Stream Ordering:

Utility:

- Stream ordering helpfully discretizes a network.
- Goal: understand network architecture
Stream Ordering:

Utility:

- Stream ordering helpfully discretizes a network.
- Goal: understand network architecture
Stream Ordering:

Utility:

- Stream ordering helpfully discretizes a network.
- Goal: understand network architecture
Stream Ordering:

Resultant definitions:

- A basin of order $\Omega$ has $n_\omega$ streams (or sub-basins) of order $\omega$.
  - $n_\omega > n_{\omega+1}$
- An order $\omega$ basin has area $a_\omega$.
- An order $\omega$ basin has a main stream length $\ell_\omega$.
- An order $\omega$ basin has a stream segment length $s_\omega$.
  1. An order $\omega$ stream segment is only that part of the stream which is actually of order $\omega$.
  2. An order $\omega$ stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams.
Stream Ordering:

Resultant definitions:

- A basin of order $\Omega$ has $n_\omega$ streams (or sub-basins) of order $\omega$.
  - $n_\omega > n_{\omega+1}$
- An order $\omega$ basin has area $a_\omega$.
- An order $\omega$ basin has a main stream length $\ell_\omega$.
- An order $\omega$ basin has a stream segment length $s_\omega$:
  1. an order $\omega$ stream segment is only that part of the stream which is actually of order $\omega$.
  2. an order $\omega$ stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams.
Stream Ordering:

Resultant definitions:

- A basin of order $\Omega$ has $n_\omega$ streams (or sub-basins) of order $\omega$.
  - $n_\omega > n_{\omega+1}$
- An order $\omega$ basin has area $a_\omega$.
- An order $\omega$ basin has a main stream length $\ell_\omega$.
- An order $\omega$ basin has a stream segment length $s_\omega$.
  1. an order $\omega$ stream segment is only that part of the stream which is actually of order $\omega$.
  2. an order $\omega$ stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams.
Stream Ordering:

Resultant definitions:

- A basin of order $\Omega$ has $n_\omega$ streams (or sub-basins) of order $\omega$.
  - $n_\omega > n_{\omega+1}$
- An order $\omega$ basin has area $a_\omega$.
- An order $\omega$ basin has a main stream length $\ell_\omega$.
- An order $\omega$ basin has a stream segment length $s_\omega$.
  1. An order $\omega$ stream segment is only that part of the stream which is actually of order $\omega$.
  2. An order $\omega$ stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams.
Stream Ordering:

Resultant definitions:

- A basin of order $\Omega$ has $n_\omega$ streams (or sub-basins) of order $\omega$.
  - $n_\omega > n_{\omega+1}$
- An order $\omega$ basin has area $a_\omega$.
- An order $\omega$ basin has a main stream length $\ell_\omega$.
- An order $\omega$ basin has a stream segment length $s_\omega$
  1. An order $\omega$ stream segment is only that part of the stream which is actually of order $\omega$
  2. An order $\omega$ stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams
Stream Ordering:

Resultant definitions:

- A basin of order $\Omega$ has $n_\omega$ streams (or sub-basins) of order $\omega$.
  - $n_\omega > n_{\omega+1}$
- An order $\omega$ basin has area $a_\omega$.
- An order $\omega$ basin has a main stream length $\ell_\omega$.
- An order $\omega$ basin has a stream segment length $s_\omega$
  1. an order $\omega$ stream segment is only that part of the stream which is actually of order $\omega$
  2. an order $\omega$ stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams
Stream Ordering:

Resultant definitions:

- A basin of order $\Omega$ has $n_\omega$ streams (or sub-basins) of order $\omega$.
  - $n_\omega > n_{\omega+1}$
- An order $\omega$ basin has area $a_\omega$.
- An order $\omega$ basin has a main stream length $\ell_\omega$.
- An order $\omega$ basin has a stream segment length $s_\omega$
  1. an order $\omega$ stream segment is only that part of the stream which is actually of order $\omega$
  2. an order $\omega$ stream segment runs from the basin outlet up to the junction of two order $\omega - 1$ streams
Horton’s laws

Self-similarity of river networks

- First quantified by Horton (1945)\textsuperscript{[3]}, expanded by Schumm (1956)\textsuperscript{[5]}

Three laws:

- Horton’s law of stream numbers:
  \[ n_\omega / n_{\omega+1} = R_n > 1 \]

- Horton’s law of stream lengths:
  \[ \bar{\ell}_{\omega+1} / \bar{\ell}_\omega = R_\ell > 1 \]

- Horton’s law of basin areas:
  \[ \bar{a}_{\omega+1} / \bar{a}_\omega = R_a > 1 \]
Horton’s laws

Self-similarity of river networks

- First quantified by Horton (1945) \(^3\), expanded by Schumm (1956) \(^5\)

Three laws:

- Horton’s law of stream numbers:
  \[
  \frac{n_\omega}{n_{\omega+1}} = R_n > 1
  \]

- Horton’s law of stream lengths:
  \[
  \frac{\bar{\ell}_{\omega+1}}{\bar{\ell}_\omega} = R_\ell > 1
  \]

- Horton’s law of basin areas:
  \[
  \frac{\bar{a}_{\omega+1}}{\bar{a}_\omega} = R_a > 1
  \]
Horton’s laws

Self-similarity of river networks

- First quantified by Horton (1945)\(^3\), expanded by Schumm (1956)\(^5\)

Three laws:

- Horton’s law of stream numbers:
  \[
  \frac{n_\omega}{n_{\omega+1}} = R_n > 1
  \]

- Horton’s law of stream lengths:
  \[
  \frac{\bar{\ell}_{\omega+1}}{\bar{\ell}_\omega} = R_\ell > 1
  \]

- Horton’s law of basin areas:
  \[
  \frac{\bar{a}_{\omega+1}}{\bar{a}_\omega} = R_a > 1
  \]
Horton’s laws

Self-similarity of river networks

First quantified by Horton (1945) [3], expanded by Schumm (1956) [5]

Three laws:

- Horton’s law of stream numbers:
  \[ \frac{n_\omega}{n_{\omega+1}} = R_n > 1 \]

- Horton’s law of stream lengths:
  \[ \frac{\bar{\ell}_{\omega+1}}{\bar{\ell}_\omega} = R_\ell > 1 \]

- Horton’s law of basin areas:
  \[ \frac{\bar{a}_{\omega+1}}{\bar{a}_\omega} = R_a > 1 \]
Horton’s laws
Self-similarity of river networks

- First quantified by Horton (1945) [3], expanded by Schumm (1956) [5]

Three laws:

- Horton’s law of stream numbers:
  \[ \frac{n_\omega}{n_{\omega+1}} = R_n > 1 \]

- Horton’s law of stream lengths:
  \[ \frac{\bar{l}_{\omega+1}}{\bar{l}_\omega} = R_\ell > 1 \]

- Horton’s law of basin areas:
  \[ \frac{\bar{a}_{\omega+1}}{\bar{a}_\omega} = R_a > 1 \]
Horton’s laws

Self-similarity of river networks

▪ First quantified by Horton (1945) \(^3\), expanded by Schumm (1956) \(^5\)

Three laws:

▪ Horton’s law of stream numbers:

\[
\frac{n_\omega}{n_{\omega+1}} = R_n > 1
\]

▪ Horton’s law of stream lengths:

\[
\frac{\bar{\ell}_{\omega+1}}{\bar{\ell}_\omega} = R_\ell > 1
\]

▪ Horton’s law of basin areas:

\[
\frac{\bar{a}_{\omega+1}}{\bar{a}_\omega} = R_a > 1
\]
Horton’s laws

Horton’s Ratios:

So... Horton’s laws are defined by three ratios: $R_n$, $R_\ell$, and $R_a$.

Horton’s laws describe exponential decay or growth:

\[
\begin{align*}
n_\omega &= n_{\omega-1} / R_n \\
&= n_{\omega-2} / R_n^2 \\
&\vdots \\
&= n_1 / R_n^{\omega-1} \\
&= n_1 e^{-(\omega-1) \ln R_n}
\end{align*}
\]
Horton’s laws

Horton’s Ratios:

- So... Horton’s laws are defined by three ratios: $R_n$, $R_\ell$, and $R_a$.

- Horton’s laws describe exponential decay or growth:

$$n_\omega = n_{\omega - 1}/R_n$$

$$= n_{\omega - 2}/R_n^2$$

$$\vdots$$

$$= n_1/R_n^{\omega - 1}$$

$$= n_1 e^{-(\omega - 1) \ln R_n}$$
Horton’s laws

Horton’s Ratios:

- So... Horton’s laws are defined by three ratios:
  
  \[ R_n, R_\ell, \text{ and } R_a. \]

- Horton’s laws describe exponential decay or growth:
  
  \[
  n_\omega = \frac{n_{\omega-1}}{R_n} \\
  = \frac{n_{\omega-2}}{R_n^2} \\
  \vdots \\
  = \frac{n_1}{R_n^{\omega-1}} \\
  = n_1 e^{-(\omega-1) \ln R_n}
  \]
Horton’s laws

Horton’s Ratios:

▷ So... Horton’s laws are defined by three ratios:

\[ R_n, \ R_\ell, \ \text{and} \ R_a. \]

▷ Horton’s laws describe **exponential decay or growth**:

\[
\begin{align*}
  n_\omega &= n_{\omega-1} / R_n \\
  &= n_{\omega-2} / R_n^2 \\
  &\vdots \\
  &= n_1 / R_n^{\omega-1} \\
  &= n_1 e^{- (\omega-1) \ln R_n}
\end{align*}
\]
Horton’s laws

Horton’s Ratios:

- So... Horton’s laws are defined by three ratios: $R_n$, $R_\ell$, and $R_a$.

- Horton’s laws describe exponential decay or growth:

\[
\begin{align*}
n_\omega &= n_{\omega-1}/R_n \\
&= n_{\omega-2}/R_n^2 \\
&= n_{\omega-3}/R_n^3 \\
&\vdots \\
&= n_1/R_n^{\omega-1} \\
&= n_1 e^{-(\omega-1) \ln R_n}
\end{align*}
\]
Horton’s laws

Similar story for area and length:

\[ \bar{a}_\omega = \bar{a}_1 e^{(\omega - 1) \ln R_a} \]

\[ \bar{\ell}_\omega = \bar{\ell}_1 e^{(\omega - 1) \ln R_\ell} \]

As stream order increases, number drops and area and length increase.
Horton’s laws

Similar story for area and length:

- \( \bar{a}_\omega = \bar{a}_1 e^{(\omega - 1) \ln R_a} \)
- \( \bar{l}_\omega = \bar{l}_1 e^{(\omega - 1) \ln R_l} \)

As stream order increases, number drops and area and length increase.
Horton’s laws

Similar story for area and length:

\[ \bar{a}_\omega = \bar{a}_1 e^{(\omega-1) \ln R_a} \]

\[ \bar{\ell}_\omega = \bar{\ell}_1 e^{(\omega-1) \ln R_\ell} \]

As stream order increases, number drops and area and length increase.
Horton’s laws

A few more things:

- Horton’s laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton’s ratios go a long way to defining a branching network...
- But we need one other piece of information...
Horton’s laws

A few more things:

- Horton’s laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton’s ratios go a long way to defining a branching network...
- But we need one other piece of information...
Horton’s laws

A few more things:

- Horton’s laws are laws of averages.
- Averaging for number is **across** basins.
- Averaging for stream lengths and areas is **within** basins.
- Horton’s ratios go a long way to defining a branching network...
- But we need one other piece of information...
Horton’s laws

A few more things:

- Horton’s laws are laws of averages.
- Averaging for number is **across** basins.
- Averaging for stream lengths and areas is **within** basins.
- Horton’s ratios go a long way to defining a branching network...
- But we need one other piece of information...
Horton’s laws

A few more things:

- Horton’s laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton’s ratios go a long way to defining a branching network...
- But we need one other piece of information...
Horton’s laws

A few more things:

- Horton’s laws are laws of averages.
- Averaging for number is across basins.
- Averaging for stream lengths and areas is within basins.
- Horton’s ratios go a long way to defining a branching network...
- But we need one other piece of information...
Horton’s laws

A bonus law:

- Horton’s law of stream segment lengths:
  \[
  \frac{\bar{s}_{\omega+1}}{\bar{s}_\omega} = R_s > 1
  \]

- Can show that \( R_s = R_\ell \).
Horton’s laws

A bonus law:

- Horton’s law of stream segment lengths:

\[ \frac{s_{\omega+1}}{s_\omega} = R_s > 1 \]

- Can show that \( R_s = R_\ell \).
Horton’s laws in the real world:

The Mississippi

The Nile

The Amazon
Horton’s laws-at-large

Blood networks:
- Horton’s laws hold for sections of cardiovascular networks
- Measuring such networks is tricky and messy...
- Vessel diameters obey an analogous Horton’s law.
Horton’s laws-at-large

Blood networks:
- Horton’s laws hold for sections of cardiovascular networks
- Measuring such networks is tricky and messy...
- Vessel diameters obey an analogous Horton’s law.
Horton’s laws-at-large

Blood networks:

- Horton’s laws hold for sections of cardiovascular networks
- Measuring such networks is tricky and messy...
- Vessel diameters obey an analogous Horton’s law.
Horton’s laws-at-large

Blood networks:

- Horton’s laws hold for sections of cardiovascular networks
- Measuring such networks is tricky and messy...
- Vessel diameters obey an analogous Horton’s law.
Observations:

- Horton’s ratios vary:
  - $R_n$: 3.0–5.0
  - $R_a$: 3.0–6.0
  - $R_\ell$: 1.5–3.0

- No accepted explanation for these values.

- Horton’s laws tell us how quantities vary from level to level ...

- ... but they don’t explain how networks are structured.
Horton’s laws

Observations:

- Horton’s ratios vary:
  - $R_n$: 3.0–5.0
  - $R_a$: 3.0–6.0
  - $R_\ell$: 1.5–3.0

- No accepted explanation for these values.

- Horton’s laws tell us how quantities vary from level to level ...

- ... but they don’t explain how networks are structured.
Horton’s laws

Observations:

- Horton’s ratios vary:
  
  \[
  \begin{array}{|c|c|}
  \hline
  \text{} & 3.0–5.0 \\
  \hline
  R_n & 3.0–5.0 \\
  R_a & 3.0–6.0 \\
  R_\ell & 1.5–3.0 \\
  \hline
  \end{array}
  \]

- No accepted explanation for these values.
- Horton’s laws tell us how quantities vary from level to level ...
- ... but they don’t explain how networks are structured.
Horton’s laws

Observations:

- Horton’s ratios vary:
  
  \[
  \begin{align*}
  R_n &\quad 3.0–5.0 \\
  R_a &\quad 3.0–6.0 \\
  R_\ell &\quad 1.5–3.0 
  \end{align*}
  \]

- No accepted explanation for these values.

- Horton’s laws tell us how quantities vary from level to level ...

- ... but they don’t explain how networks are structured.
Tokunaga’s law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure [7, 8, 9]
- As per Horton-Strahler, use stream ordering.
- **Focus**: describe how streams of different orders connect to each other.
- Tokunaga’s law is also a law of averages.
Tokunaga’s law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure $[7, 8, 9]$
- As per Horton-Strahler, use stream ordering.
- Focus: describe how streams of different orders connect to each other.
- Tokunaga’s law is also a law of averages.
Tokunaga’s law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure [7, 8, 9]
- As per Horton-Strahler, use stream ordering.
  - Focus: describe how streams of different orders connect to each other.
- Tokunaga’s law is also a law of averages.
Tokunaga’s law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure \([7, 8, 9]\).
- As per Horton-Strahler, use stream ordering.
- **Focus**: describe how streams of different orders connect to each other.
- Tokunaga’s law is also a law of averages.
Tokunaga’s law

Delving deeper into network architecture:

- Tokunaga (1968) identified a clearer picture of network structure [7, 8, 9]
- As per Horton-Strahler, use *stream ordering*.
- **Focus:** describe how streams of different orders connect to each other.
- Tokunaga’s law is also a law of averages.
Network Architecture

Definition:

- $T_{\mu,\nu} = \text{the average number of side streams of order } \nu \text{ that enter as tributaries to streams of order } \mu$
- $\mu, \nu = 1, 2, 3, \ldots$
- $\mu \geq \nu + 1$
- Recall each stream segment of order $\mu$ is ‘generated’ by two streams of order $\mu - 1$
- These generating streams are not considered side streams.
Network Architecture

Definition:

- $T_{\mu,\nu} =$ the average number of side streams of order $\nu$ that enter as tributaries to streams of order $\mu$

- $\mu, \nu = 1, 2, 3, \ldots$

- $\mu \geq \nu + 1$

- Recall each stream segment of order $\mu$ is ‘generated’ by two streams of order $\mu - 1$

- These generating streams are not considered side streams.
Network Architecture

Definition:

- $T_{\mu,\nu} = \text{the average number of side streams of order } \nu \text{ that enter as tributaries to streams of order } \mu$
- $\mu, \nu = 1, 2, 3, \ldots$
- $\mu \geq \nu + 1$
- Recall each stream segment of order $\mu$ is ‘generated’ by two streams of order $\mu - 1$
- These generating streams are not considered side streams.
Network Architecture

Definition:

- \( T_{\mu,\nu} \) = the average number of side streams of order \( \nu \) that enter as tributaries to streams of order \( \mu \)
- \( \mu, \nu = 1, 2, 3, \ldots \)
- \( \mu \geq \nu + 1 \)
- Recall each stream segment of order \( \mu \) is ‘generated’ by two streams of order \( \mu - 1 \)
- These generating streams are not considered side streams.
Network Architecture

Definition:

- $T_{\mu,\nu} =$ the average number of side streams of order $\nu$ that enter as tributaries to streams of order $\mu$
- $\mu, \nu = 1, 2, 3, \ldots$
- $\mu \geq \nu + 1$
- Recall each stream segment of order $\mu$ is ‘generated’ by two streams of order $\mu - 1$
- These generating streams are not considered side streams.
Network Architecture

Tokunaga’s law

- Property 1: Scale independence—depends only on difference between orders:

\[ T_{\mu,\nu} = T_{\mu - \nu} \]

- Property 2: Number of side streams grows exponentially with difference in orders:

\[ T_{\mu,\nu} = T_1 (R_T)^{\mu - \nu - 1} \]

- We usually write Tokunaga’s law as:

\[ T_k = T_1 (R_T)^{k-1} \]

where \( R_T \approx 2 \)
Network Architecture

Tokunaga’s law

- Property 1: Scale independence—depends only on difference between orders:

\[ T_{\mu,\nu} = T_{\mu-\nu} \]

- Property 2: Number of side streams grows exponentially with difference in orders:

\[ T_{\mu,\nu} = T_1 (R_T)^{\mu-\nu-1} \]

- We usually write Tokunaga’s law as:

\[ T_k = T_1 (R_T)^{k-1} \] where \( R_T \approx 2 \)
Network Architecture

Tokunaga’s law

- Property 1: Scale independence—depends only on difference between orders:

  \[ T_{\mu,\nu} = T_{\mu-\nu} \]

- Property 2: Number of side streams grows exponentially with difference in orders:

  \[ T_{\mu,\nu} = T_1(R_T)^{\mu-\nu-1} \]

- We usually write Tokunaga’s law as:

  \[ T_k = T_1(R_T)^{k-1} \quad \text{where} \quad R_T \sim 2 \]
Network Architecture

Tokunaga’s law

➢ Property 1: Scale independence—depends only on difference between orders:

\[ T_{\mu,\nu} = T_{\mu-\nu} \]

➢ Property 2: Number of side streams grows exponentially with difference in orders:

\[ T_{\mu,\nu} = T_1(R_T)^{\mu-\nu-1} \]

➢ We usually write Tokunaga’s law as:

\[ T_k = T_1(R_T)^{k-1} \quad \text{where} \quad R_T \approx 2 \]
Network Architecture

Tokunaga’s law

- Property 1: Scale independence—depends only on difference between orders:

\[ T_{\mu,\nu} = T_{\mu-\nu} \]

- Property 2: Number of side streams grows exponentially with difference in orders:

\[ T_{\mu,\nu} = T_1(R_T)^{\mu-\nu-1} \]

- We usually write Tokunaga’s law as:

\[ T_k = T_1(R_T)^{k-1} \quad \text{where} \quad R_T \approx 2 \]
Network Architecture

Tokunaga’s law

- Property 1: Scale independence—depends only on difference between orders:

\[ T_{\mu,\nu} = T_{\mu-\nu} \]

- Property 2: Number of side streams grows exponentially with difference in orders:

\[ T_{\mu,\nu} = T_1(R_T)^{\mu-\nu-1} \]

- We usually write Tokunaga’s law as:

\[ T_k = T_1(R_T)^{k-1} \quad \text{where} \quad R_T \approx 2 \]
Tokunaga’s law—an example:

\[ T_1 \approx 2 \]

\[ R_T \approx 4 \]
The Mississippi

A Tokunaga graph:

\[ \log_{10} \langle T_{\mu,\nu} \rangle \]

\[ \nu = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \]
Nutshell:

Branching networks I:

- Show remarkable **self-similarity** over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler *Stream ordering* gives one useful way of getting at the architecture of branching networks.
- Horton’s laws reveal self-similarity.
- Horton’s laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga’s laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).
Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton’s laws reveal self-similarity.
- Horton’s laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga’s laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).
Nutshell:

Branching networks I:

- Show remarkable **self-similarity** over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler **Stream ordering** gives one useful way of getting at the architecture of branching networks.
- Horton’s laws reveal self-similarity.
- Horton’s laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga’s laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).
Nutshell:

Branching networks I:

- Show remarkable **self-similarity** over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler **Stream ordering** gives one useful way of getting at the architecture of branching networks.
- **Horton’s laws** reveal self-similarity.
- Horton’s laws can be misinterpreted as suggesting a pure hierarchy.
- **Tokunaga’s laws** neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).
Nutshell:

Branching networks I:

- Show remarkable self-similarity over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler Stream ordering gives one useful way of getting at the architecture of branching networks.
- Horton’s laws reveal self-similarity.
- Horton’s laws can be misinterpreted as suggesting a pure hierarchy.
- Tokunaga’s laws neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).
Nutshell:

Branching networks I:

- Show remarkable **self-similarity** over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler **Stream ordering** gives one useful way of getting at the architecture of branching networks.
- **Horton’s laws** reveal self-similarity.
- Horton’s laws can be misinterpreted as suggesting a pure hierarchy.
- **Tokunaga’s laws** neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).
Nutshell:

Branching networks I:

- Show remarkable **self-similarity** over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler **Stream ordering** gives one useful way of getting at the architecture of branching networks.
- **Horton’s laws** reveal self-similarity.
- Horton’s laws can be misinterpreted as suggesting a pure hierarchy.
- **Tokunaga’s laws** neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).
Nutshell:

Branching networks I:

- Show remarkable **self-similarity** over many scales.
- There are many interrelated scaling laws.
- Horton-Strahler **Stream ordering** gives one useful way of getting at the architecture of branching networks.
- **Horton’s laws** reveal self-similarity.
- Horton’s laws can be misinterpreted as suggesting a pure hierarchy.
- **Tokunaga’s laws** neatly describe network architecture.
- Branching networks exhibit a mixed hierarchical structure.
- Horton and Tokunaga can be connected analytically (next up).
References


References II

*Fractal River Basins: Chance and Self-Organization.*  

Evolution of drainage systems and slopes in badlands at Perth Amboy, New Jersey.  

Hypsometric (area altitude) analysis of erosional topography.  
References III

